



INSTITUTO POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO
Departamento de Engenharia Informática

Provisionamento em Infraestruturas de Voz Sobre IP

Paulo Alexandre Pereira Gomes

Fevereiro 2011



INSTITUTO POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO
Departamento de Engenharia Informática

Provisionamento em Infraestruturas de Voz Sobre IP

Dissertação submetida ao Instituto Politécnico de Leiria para
obtenção do grau de Mestre em Engenharia Informática e
Computação Móvel

Paulo Alexandre Pereira Gomes

Orientada pelo Professor Doutor Carlos Rabadão

Fevereiro 2011

“A Liberdade é podermos traçar o nosso próprio limite”

Ricardo Esteves

Agradecimentos

Quero aqui expressar os meus sinceros agradecimentos às pessoas e instituições que, directa e indirectamente, contribuíram para a realização deste trabalho.

Começo com o meu imenso agradecimento ao Professor Doutor Carlos Rabadão pela sua enorme disponibilidade, pela sua orientação e pelo seu incentivo, prestados ao longo das diversas fases deste trabalho, sem os quais teria sido impossível concluí-lo com sucesso.

Agradeço também aos Serviços Informáticos do Instituto Politécnico de Leiria pelos meios e condições que colocaram ao meu dispor para a realização deste trabalho.

Agradeço ainda ao Instituto Politécnico de Leiria e à Escola Superior de Tecnologia e Gestão de Leiria, pelas condições que colocaram ao meu dispor para a realização deste trabalho.

Por fim, quero expressar todo o meu agradecimento pelo apoio e compreensão que recebi da minha esposa Rita, da minha família e amigos.

Resumo

Actualmente vive-se em plena era digital. Redes sociais, chamadas telefónicas pela Internet, telefones com vídeo-chamada, telemóveis e televisões com acesso à Internet, são tecnologias que não se podem ignorar. Assumem-se hoje como novas formas de comunicar, que aliam facilidade de utilização, maior alcance e custos mais reduzidos.

Esta dissertação foca-se na tecnologia de telefonia utilizando uma rede de comutação de pacotes, como por exemplo, a Internet. Esta tecnologia denominada de voz sobre protocolo de Internet (VoIP), está hoje em grande expansão nas empresas e instituições devido, em parte, à redução de custos e aos novos serviços suportados.

O aparecimento da telefonia IP trouxe inúmeras vantagens para o utilizador. Contudo, existe uma problemática associada, que são os custos de suporte e manutenção que esta tecnologia carece. Em tecnologias anteriores, os equipamentos telefónicos não necessitavam de configurações significativas, localmente em cada dispositivo, sendo praticamente tudo parametrizado na central telefónica. O mesmo não acontece em sistemas de VoIP.

O Instituto Politécnico de Leiria (IPL) conta já com cerca de 300 telefones IP em produção e sempre que é preciso fazer algum tipo de manutenção em massa esta constitui-se uma actividade demorada, envolvendo muita mão-de-obra. Assim, neste projecto é proposta uma arquitectura que permita gerir todos estes equipamentos, em simultâneo, tendo em conta as funcionalidades de provisionamento suportadas por cada um.

A arquitectura proposta baseia-se essencialmente em cinco fases – (i) provisionamento de um novo terminal, (ii) verificação de terminais registados no Asterisk, (iii) visualização de terminais adicionados à plataforma, (iv) alterações de configurações em massa, (v) actualização de *firmware* e (vi) tolerância a falhas.

Para proceder à validação desta arquitectura foi desenvolvido um protótipo, tendo-se posteriormente efectuado a análise e avaliação do modelo proposto. Com esta respectiva análise e avaliação foi possível validar a arquitectura relativamente à sua utilidade para reduzir tempos em suporte e manutenção do sistema de telefonia IP do IPL.

.

Abstract

Nowadays we are living in the digital age. Social networks, internet phone calls, phones with video calling, mobile phones and televisions with internet access, are technologies that can't be ignored. Today, those are assumed new forms of communication, combining ease of use, greater reach and lower costs.

This dissertation focuses on telephony technology using a packet switching network, such as, the internet.

This technology called Voice over Internet Protocol (VoIP) is now booming in enterprises and institutions, due in part to reduced costs and new services supported.

The emergence of IP telephony has brought many benefits for users. However, there is an associated problem, which is the cost of maintenance and support that this technology is needed. In previous technologies, telephone equipment didn't required significant configuration, locally on each device, being practically everything parameterized in the switchboard. The same don't happen in VoIP systems.

The Instituto Politécnico de Leiria (IPL) already has about 300 IP phones in production and whenever need to do some kind of mass maintenance this constitutes a time consuming activity, involving a lot of manpower. Thus, in this project is propose an architecture that allows manage all these equipments, simultaneously, taking into account the provisioning features supported by each.

The proposed architecture is mainly based on five stages – (i) provisioning of a new terminal, (ii) - verification of the registered terminals, (iii) – visualization of the

terminals added to the platform, (iv) – changes of the configuration in mass, (v) – firmware update and (vi) - fault tolerance.

To proceed with the validation of this architecture a prototype was developed, and it was subsequently performed the analysis and evaluation of the proposed model. With this analysis and its evaluation was possible to validate the architecture relatively to its effect to reduce time on support and maintenance of the IP telephony system of IPL.

Índice

AGRADECIMENTOS.....	7
RESUMO.....	VIII
ABSTRACT.....	X
ÍNDICE.....	XIII
LISTA DE FIGURAS.....	XVI
LISTA DE TABELAS.....	XIX
LISTA DE ACRÓNIMOS.....	XX
1 INTRODUÇÃO.....	1
1.1 O PROJECTO VoIP@RCTS	4
1.2 MOTIVAÇÃO E OBJECTIVOS.....	5
1.3 RESUMO DAS PRINCIPAIS CONTRIBUIÇÕES.....	6
1.4 ESTRUTURA DA DISSERTAÇÃO.....	6
2 ESTADO DA ARTE.....	9
2.1 VOZ SOBRE IP.....	9
2.1.1 Compressão e codificação.....	11
2.1.2 Protocolos de sinalização.....	13
2.2 DISPOSITIVOS E EQUIPAMENTOS VoIP.....	18
2.2.1 Telefones IP (Hardphones).....	18
2.2.2 Telefones virtuais (Softphones).....	20
2.2.3 Adaptador ATA.....	23
2.2.4 Telefones de conferência.....	24
2.3 PROVISIONAMENTO.....	24
2.3.1 Asterisk.....	26
2.3.2 FreePBX.....	29
2.3.3 Voice Pulse Global Provisioning System (VGPS).....	31
2.3.4 Soluções comerciais.....	31
2.4 TÉCNICAS DE PROVISIONAMENTO.....	33
2.4.1 TFTP.....	33
2.4.2 FTP.....	34
2.4.3 HTTP	35
2.4.4 DHCP.....	36
2.5 TR-069 CPE WAN MANAGEMENT PROTOCOL.....	39
2.5.1 Síntese.....	42
3 TERMINAIS VOIP.....	44

3.1	POLYCOM SOUNDPOINT IP330.....	44
3.1.1	Personalização do idioma.....	50
3.1.2	Personalização do toque.....	51
3.1.3	Personalização da imagem de logotipo.....	52
3.1.4	Agenda de contactos.....	53
3.2	SNOM 360.....	54
3.2.1	Actualização do firmware para versão 7.....	55
3.2.2	Correio de voz.....	56
3.2.3	Personalização das teclas de atalhos.....	57
3.2.4	Personalização da imagem/logotipo.....	58
3.2.5	Carregar a agenda de contactos.....	59
3.2.6	Personalização do toque.....	60
3.2.7	Personalização do idioma.....	61
3.3	LINKSYS SPA942.....	62
3.3.1	Personalização do toque.....	64
3.3.2	Personalização da imagem/logotipo.....	65
3.3.3	Personalização do idioma.....	66
3.3.4	Personalização da agenda.....	67
3.3.5	Personalização das teclas de marcação rápida.....	69
3.4	GRANDSTREAM BT200.....	70
3.4.1	Personalização do toque.....	75
3.5	X-LITE.....	76
3.5.1	Síntese.....	77
4	ARQUITECTURA.....	79
4.1	REQUISITOS.....	79
4.2	MODELO DE ARQUITECTURA.....	81
4.3	FASES DE PROVISIONAMENTO.....	82
4.4	PROVISIONAMENTO DE UM NOVO TERMINAL.....	84
4.5	TERMINAIS REGISTADOS NO ÁSTERISK.....	85
4.6	TERMINAIS ADICIONADOS À PLATAFORMA.....	87
4.7	CONFIGURAÇÕES EM MASSA.....	88
4.8	SINCRONIZAÇÃO DOS CONTACTOS COM A AD.....	89
4.9	TOLERÂNCIA A FALHAS.....	90
5	PROTÓTIPO.....	93
5.1	CENÁRIO.....	94
5.2	IDEALIZAÇÃO DA INTERFACE.....	96
5.3	GRUPO "PLATAFORMA".....	98
5.3.1	Função "Estado".....	98
5.3.2	Função "Configurações".....	99
5.3.3	Função "Cópias de segurança".....	99
5.3.4	Função "Relatórios".....	100
5.3.5	Função "Enviar Ficheiros".....	101
5.4	GRUPO "TERMINAIS".....	102
5.4.1	Função "Adicionar".....	102
5.4.2	Função "Adicionar em massa".....	103
5.4.3	Função "Adicionados".....	104
5.4.4	Função "Registados".....	104
5.4.5	Função "Softphones".....	105
5.4.6	Função "Agenda".....	105
5.5	GRUPO "EXTENSÕES".....	107
5.5.1	Função "Detalhes".....	107
5.5.2	Função "Reiniciar".....	108
5.5.3	Função "Monitor".....	108
5.6	GRUPO "CONFIGURAÇÕES EM MASSA".....	109
5.6.1	Função "Polycom 330".....	109
5.7	PLANO DE TESTES.....	110

5.7.1 Testes de carga.....	112
5.7.2 Testes de Funcionamento.....	116
5.8 SÍNTESE.....	119
6 CONCLUSÕES.....	121
6.1 PRINCIPAIS CONTRIBUIÇÕES.....	122
6.2 TÓPICOS PARA TRABALHO FUTURO.....	123
REFERÊNCIAS.....	125
ANEXOS.....	129

Lista de Figuras

FIGURA 2.1 - CONVERSÃO VOZ/PACOTES DE DADOS.....	10
FIGURA 2.2 - EXEMPLO DE UMA LIGAÇÃO TRUNK.....	17
FIGURA 2.3 - EXEMPLO DE UM HARDPHONE.....	20
FIGURA 2.4 - TELEFONE VIRTUAL IP X-LITE.....	21
FIGURA 2.5 - SOFTPHONE OU TELEFONE VIRTUAL IP EKIGA.....	22
FIGURA 2.6 - LIGAÇÕES DISPONÍVEIS NUM ADAPTADOR ATA.....	23
FIGURA 2.7 - TELEFONE DE CONFERÊNCIA POLYCOM.....	24
FIGURA 2.8 - INTEROPERABILIDADE DO ASTERISK.....	29
FIGURA 2.9 - DIFERENÇA ENTRE TRIXBOX E FREEPBX.....	29
FIGURA 2.10 - FASES DO PROVISIONAMENTO.....	37
FIGURA 2.11 - EXEMPLO DE CONFIGURAÇÃO DE UM SERVIDOR DHCP.....	38
FIGURA 2.12 - AMBIENTE DE GESTÃO UTILIZANDO O PROTOCOLO CWMP.....	40
FIGURA 2.13 - ALGUNS PROTOCOLOS INCORPORADOS NO TR-069.....	42
FIGURA 3.1 - TERMINAL POLYCOM SOUNDPOINT IP330.....	45
FIGURA 3.2 - ÁRVORE DE DIRECTÓRIOS PARA O POLYCOM 330.....	46
FIGURA 3.3 - FASE DE ARRANQUE DO POLYCOM 330.....	47
FIGURA 3.4 - CONTEÚDO DO FICHEIRO 000000000000.CFG.....	48
FIGURA 3.5 - CONTEÚDO DO FICHEIRO 0004F200106C.CFG.....	48
FIGURA 3.6 - FICHEIRO DE CONFIGURAÇÃO DA EXTENSÃO.....	49
FIGURA 3.7 - CONTEÚDO DO FICHEIRO APP_CONFIG_330.CFG.....	50
FIGURA 3.8 - CONTEÚDO ACRESCENTADO AO FICHEIRO APP_CONFIG_330.CFG.....	51
FIGURA 3.9 - CONTEÚDO ACRESCENTADO AO FICHEIRO APP_CONFIG_330.CFG.....	52
FIGURA 3.10 - CONTEÚDO ACRESCENTADO AO FICHEIRO APP_CONFIG_330.CFG.....	52
FIGURA 3.11 - FICHEIRO DA AGENDA DE CONTACTOS.....	53
FIGURA 3.12 - TELEFONE SNOM 360.....	54
FIGURA 3.13 - FIRMWARE DO TELEFONE.....	56
FIGURA 3.14 - PROGRAMAÇÃO DA TECLA DE CORREIO DE VOZ.....	56

FIGURA 3.15 - CONFIGURAÇÃO DAS TECLAS DE ATALHO.....	57
FIGURA 3.16 - TELEFONE SNOM 360 COM MÓDULO DE EXPANSÃO.....	58
FIGURA 3.17 - PERSONALIZAÇÃO DA IMAGEM DE LOGOTIPO.....	59
FIGURA 3.18 - PERSONALIZAR A AGENDA DE CONTACTOS.....	59
FIGURA 3.19 - PERSONALIZAÇÃO DO TOQUE.....	60
FIGURA 3.20 - PERSONALIZAÇÃO DO IDIOMA.....	61
FIGURA 3.21 - PERSONALIZAÇÃO DO IDIOMA PADRÃO.....	62
FIGURA 3.22 - TELEFONE IP LINKSYS SPA942.....	62
FIGURA 3.23 - FICHEIRO INICIAL DE CONFIGURAÇÃO.....	63
FIGURA 3.24 - CONFIGURAÇÕES INDIVIDUAIS DE CADA TELEFONE.....	64
FIGURA 3.25 - PERSONALIZAÇÃO DO TOQUE DE CHAMADA.....	64
FIGURA 3.26 - PERSONALIZAÇÃO DA IMAGEM DE LOGOTIPO.....	65
FIGURA 3.27 - PERSONALIZAÇÃO DO IDIOMA.....	66
FIGURA 3.28 - AGENDA PESSOAL.....	67
FIGURA 3.29 - COMANDO PARA ENVIAR CONTACTOS PARA O SPA942.....	68
FIGURA 3.30 - CÓDIGO FONTE.....	68
FIGURA 3.31 - SPA 942 PERSONAL DIRECTORY MANAGER.....	69
FIGURA 3.32 - TECLAS DE MARCAÇÃO RÁPIDA.....	70
FIGURA 3.33 - CONFIGURAÇÃO DAS TECLAS DE MARCAÇÃO RÁPIDA.....	70
FIGURA 3.34 - TELEFONE IP GRANDSTREAM BT200.....	71
FIGURA 3.35 - DEFINIÇÕES INICIAIS.....	71
FIGURA 3.36 - CONTEÚDO DO FICHEIRO DE DEFINIÇÕES INICIAIS.....	72
FIGURA 3.37 - PARÂMETROS ADICIONAIS DO DEFINICOES_INICIAIS.CFG.....	73
FIGURA 3.38 - CONFIGURAÇÕES DE REGISTO DE CADA TELEFONE.....	74
FIGURA 3.39 - COMANDO PARA CRIAR O FICHEIRO DE CONFIGURAÇÃO DO TELEFONE.....	74
FIGURA 3.40 - COMANDO PARA TRANSFERIR O FICHEIRO DE CONFIGURAÇÃO PARA O TFTP.....	75
FIGURA 3.41 - CONVERSÃO DO FICHEIRO AUDIO PARA TOQUE DE CHAMADA.....	75
FIGURA 3.42 - COMANDO PARA TRANSFERIR O FICHEIRO PARA O TFTP.....	75
FIGURA 4.1 - MODELO DE ARQUITECTURA.....	81
FIGURA 4.2 - FASES DE ARRANQUE DE UM TERMINAL IP.....	83
FIGURA 4.3 - PROVISIONAR UM NOVO TERMINAL.....	85
FIGURA 4.4 - DIAGRAMA DE INTEROPERABILIDADE USANDO SCP.....	86
FIGURA 4.5 - DIAGRAMA DE INTEROPERABILIDADE USANDO HTTP.....	87
FIGURA 4.6 - PROCESSO PARA LISTAR TERMINAIS.....	88
FIGURA 4.7 - PROCESSO PARA ALTERAÇÃO DE CONFIGURAÇÕES EM MASSA.....	89
FIGURA 4.8 - PROCESSO DE SINCRONIZAÇÃO DOS CONTACTOS COM A AD.....	90
FIGURA 4.9 - ESQUEMA DE TOLERÂNCIA A FALHAS.....	91

FIGURA 4.10 - REPLICAÇÃO DOS DADOS DE PROVISIONAMENTO.....	92
FIGURA 5.1 - CENÁRIO UTILIZADO EM LABORATÓRIO.....	95
FIGURA 5.2 - ORGANIZAÇÃO DAS FUNCIONALIDADES NA INTERFACE WEB.....	97
FIGURA 5.3 - PÁGINA INICIAL DA PLATAFORMA DE PROVISIONAMENTO.....	97
FIGURA 5.4 - FUNÇÃO ESTADO.....	98
FIGURA 5.5 - CONFIGURAÇÕES DA PLATAFORMA.....	99
FIGURA 5.6 - CÓPIAS DE SEGURANÇA DA PLATAFORMA.....	100
FIGURA 5.7 - RELATÓRIOS DO SISTEMA.....	101
FIGURA 5.8 - EXEMPLO DE UMA MENSAGEM DE ALERTA.....	101
FIGURA 5.9 - FUNÇÃO "ENVIAR FICHEIROS".....	102
FIGURA 5.10 - ADICIONAR UM TERMINAL IP À PLATAFORMA.....	103
FIGURA 5.11 - ADICIONAR TERMINAIS EM MASSA.....	103
FIGURA 5.12 - TERMINAIS ADICIONADOS À PLATAFORMA.....	104
FIGURA 5.13 - TERMINAIS TELEFÓNICOS IP REGISTADOS.....	105
FIGURA 5.14 - SOFTPHONES.....	105
FIGURA 5.15 - SINCRONIZAÇÃO DA AGENDA DE CONTACTOS.....	106
FIGURA 5.16 - DETALHES DE UMA EXTENSÃO.....	107
FIGURA 5.17 - REINICIAR TERMINAIS IP.....	108
FIGURA 5.18 - MONITOR DE EXTENSÕES.....	109
FIGURA 5.19 - CONFIGURAÇÕES EM MASSA.....	110
FIGURA 5.20 - TESTE DE CARGA AO DHCP.....	113
FIGURA 5.21 - DESEMPENHO DA INTERFACE DE REDE.....	113
FIGURA 5.22 - DESEMPENHO DO PROCESSADOR.....	114
FIGURA 5.23 - DESEMPENHO DA MEMÓRIA.....	114
FIGURA 5.24 - SCRIPT UDP_STRESS_CLIENT.PY.....	115
FIGURA 5.25 - DESEMPENHO DA INTERFACE DE REDE.....	115
FIGURA 5.26 - DESEMPENHO DO PROCESSADOR.....	116
FIGURA 5.27 - DESEMPENHO DE MEMÓRIA.....	116
FIGURA 1 - GRANDSTREAM CONFIGURATION GENERATOR.....	130
FIGURA 2 - RINGTOOL.....	130
FIGURA 3 - AUTOIT WINDOWS INFO.....	132
FIGURA 4 - CÓDIGO PARA EXECUTAR O INSTALADOR.....	133
FIGURA 5 - INTERACÇÃO DO "FINDER TOOL" COM UM RADIO BUTTON.....	133
FIGURA 6 - ECRÃ INICIAL.....	134
FIGURA 7 - SELECÇÃO DO ACORDO DE LICENÇA.....	134
FIGURA 8 - ACEITAÇÃO DO ACORDO DA LICENÇA.....	135
FIGURA 9 - BOTÃO "NEXT" ATÉ AO FINAL DA INSTALAÇÃO.....	135

FIGURA 10 - JANELA DE CONFIGURAÇÃO DA CONTA SIP NO X-LITE.....	136
FIGURA 11 - CONFIGURAÇÃO DA CONTA SIP.....	136
FIGURA 12 - PROPRIEDADES DA CONTA SIP.....	136
FIGURA 13 - INTRODUÇÃO DE DADOS DE AUTENTICAÇÃO DA CONTA SIP.....	137
FIGURA 14 - ECRÃ FINAL DA INSTALAÇÃO.....	138
FIGURA 15 - INTERFACE DO AUTOIT SCRIPT TO EXE CONVERTER.....	138

Lista de Tabelas

TABELA 2.1 - CODECS [SAHABUDIN09][MINOLI98][RODMAN08][MACIAN08][GIBSON05].....	12
TABELA 2.2 - FUNÇÃO DOS ELEMENTOS DE REDE DO H.323.....	14
TABELA 2.3 - FUNÇÕES DOS ELEMENTOS SIP.....	16
TABELA 2.4 - TELEFONES IP EXISTENTE NO IPL.....	18
TABELA 2.5 - ALGUMAS FUNCIONALIDADES DE UMA INTERFACE WEB DO ASTERISK.....	28
TABELA 2.6 - SOLUÇÕES COMERCIAIS COM MÓDULO DE PROVISIONAMENTO.....	32
TABELA 3.1 - FUNÇÃO DOS DIRECTÓRIOS CONTIDOS EM /HOME/POLYCOM.....	46
TABELA 3.2 - RESUMO DE FUNÇÕES DO POLYCOM 330.....	50
TABELA 3.3 - ACÇÕES PARA BOTÕES DE ATALHO.....	58
TABELA 4.1 - REQUISITOS PARA SOLUÇÃO DE PROVISIONAMENTO.....	80
TABELA 5.1 - FUNCIONALIDADES DESENVOLVIDAS.....	94
TABELA 5.2 - PLANO DE TESTES AOS VÁRIOS COMPONENTES DO SERVIDOR DE PROVISIONAMENTO	111
TABELA 5.3 - TESTES DE FUNCIONAMENTO.....	112

Lista de Acrónimos

PSTN	Public Switched Telephone Network
LAN	Local Area Network
WAN	Wide Area Network
VoIP	Voice Over IP
PBX	Private Branch Exchange
IPPBX	IP Private Branch Exchange
IP	Internet Protocol
ATA	Analog Telephone Adapter
IPL	Instituto Politécnico de Leiria
FCCN	Fundação para a Computação Científica Nacional
RCTS	Rede Ciência, Tecnologia e Sociedade
TFTP	Trivial File Transfer Protocol
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
DHCP	Dynamic Host Configuration Protocol
ADSL	Asymmetric Digital Subscriber Line
TCP/IP	Transmission Control Protocol/Internet Protocol
CODEC	Coder-Decoder
PDA	Personal Digital Assistant
Wi-Fi	Wireless Fidelity
ATM	Asynchronous Transfer Mode
QoS	Quality of Service
MOS	Mean Opinion Score

SIP	Session Initiation Protocol
IAX	Inter Asterisk eXchange
REDIS	Rede Digital com Integração de Serviços
ITU-T	International Telecommunication Union Telecommunication Standardization
MGC	Media Gateway Controller
MG	Media Gateway
MCU	Multipoint Control Units
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
DNS	Domain Name System
MMUSIC	Multiparty Multimedia Session Control
IETF	Internet Engineering Task Force
RFC	Request For Comments
UA	User Agent
NAT	Network Address Translator
XML	eXtensible Markup Language
GPL	General Public License
ACD	Automatic Call Distribution
IVR	Interactive Voice Response
API	Application Programming Interface
TI	Tecnologias de Informação
DDI/DID	Direct Dial-In / Direct Inward Dialing
PC	Personal Computer
POTS	Plain Old Telephone Service
PHP	Hypertext Preprocessor
MAC	Media Access Control
VGPS	Voice Pulse Global Provisioning System
NTP	Network Time Protocol
CPE	Customer Premises Equipment
CWMP	CPE WAN Management Protocol
ACS	Auto Configuration Servers
SOAP	Simple Object Access Protocol
RPC	Remote Procedure Call)
WWW	World Wide Web
W3C	World Wide Web Consortium
SSL	Secure Sockets Layer

TLS	Transport Layer Security
WAV	Waveform Audio File Format
BMP	Bitmap
UARS	Unidade de Administração de Redes e Sistemas
SCP	Security Copy Protocol
BD	Base de Dados
SSH	Secure Shell
XMPP	Extensible Messaging and Presence Protocol
OU	Organizational Unit

1 Introdução

Nas duas últimas décadas a rede pública de serviço telefónico (PSTN) evoluiu de uma estrutura totalmente analógica para a comutação digital de circuitos. O suporte da transmissão do sinal analógico na rede digital tem de ser codificado ou convertido em formato digital antes de entrar na rede pública. Na outra extremidade o sinal é decodificado ou convertido novamente para a sua forma original, ou seja, em formato analógico. [Chong04]

Uma infraestrutura de comutação de circuitos funciona como uma ligação dedicada, isto é, uma ligação entre dois nós que não é partilhada com outras até a comunicação ser terminada.

Nos últimos anos, tem-se assistido a uma nova mudança de paradigma com o avanço da tecnologia de comutação de pacotes para transmitir voz e dados. Uma rede telefónica de comutação de pacotes foi estruturada para o intercâmbio de informação em formato digital, entre computadores ou outros dispositivos. Esta rede pode estender-se de uma área geográfica limitada (LAN), como um edifício ou um Campus, até pontos de acesso espalhados por várias cidades, ou mesmo países (WAN).[Pires06]

As redes de comutação de pacotes funcionam segmentando a informação em unidades mais pequenas (pacotes ou datagramas) de forma a serem enviadas e a poderem viajar para o destino, as quais podem viajar por diferentes caminhos, chegarem em diferentes ordens e tempos diferentes.[Pires06]

A Internet, um exemplo claro de uma rede WAN, é um dos fenómenos de globalização mais marcantes dos últimos tempos. Evolução rápida, oferta de aplicações acessíveis a nível global, pesquisa e partilha de informação em áreas de interesse muito diversas, transacções comerciais, redes sociais ou mesmo entretenimento, são alguns dos factores que tornam esta rede tão difundida a nível mundial. A Internet evoluiu da partilha de pequenos conteúdos até à difusão de produtos multimédia em tempo real. [Florin10]

A telefonia IP, também conhecida por voz sobre IP (VoIP), surgiu aproveitando as capacidades multimédia que a estrutura de uma rede de comutação de pacotes oferece, tais como, partilha do meio de comunicação, mobilidade, redução de custos, débitos de informação, partilha de recursos, entre outros.

Um dos factores que pode ter levado à grande massificação desta tecnologia foi o aparecimento de redes de banda larga, pois permitiu acessos com grandes débitos de informação face às linhas convencionais (analógicas). Este aumento de largura de banda, de acesso à Internet, tornaram possível a utilização de serviços como a comunicação de voz e vídeo até ao cliente final sem atrasos significativos. Hoje em dia já é possível telefonar gratuitamente para diversos pontos do globo.[Robert08]

A par da telefonia IP surgiram também as *Privates Branch Exchange* (PBX) baseadas em redes IP, que contribuíram para uma significativa revolução na área das telecomunicações de voz. A telefonia IP levou, entre outros aspectos, a uma considerável redução de preços das comunicações quer para o utilizador final quer para operadores. Já há muito que as chamadas telefónicas pela Internet são uma realidade, podendo estas serem de PC-para-PC, PC-para-Telefone e mesmo Telefone-para-PC.

O aparecimento do Asterisk, uma implementação livre de PBX baseado em IP (IPPBX), permitiu uma massificação deste tipo de centrais telefónicas um pouco por todo o mundo.

Devido à sua Licença Pública de uso Geral e à quantidade de serviços que oferece a custo zero, os IPPBX baseados em Asterisk, estão a ganhar mais terreno face aos PBX tradicionais.

O mercado e os utilizadores exigem cada vez mais desta tecnologia, no sentido em que reivindicam mais opções, personalizações, funcionalidades e actualizações. Daqui resulta também a necessidade dos próprios administradores precisarem de mais ferramentas para gerir todas estas infraestruturas de uma forma mais produtiva, pois a tarefa de configurar/personalizar cada terminal individualmente consome muito tempo de mão-de-obra.

A evolução desta área levou a que fabricantes de telefones IP também inovassem os seus equipamentos e, portanto, actualmente os telefones IP são dotados de variadas funcionalidades, não só para o utilizador mas também para os administradores de sistemas de voz sobre IP, com a finalidade de auxiliar e otimizar a gestão dos mesmos.

A gestão de um parque grande de telefones IP, adaptadores analógicos (ATA¹) e *softphones*, pode ser uma tarefa bastante complexa e exaustiva para o administrador, imagine-se o seguinte cenário: a necessidade de actualizar o *firmware* a 100 telefones. Se a actualização de cada *firmware* demorar em média 15 minutos, e o mesmo tempo para reconfigurar o telefone e fazer os respectivos testes, então ocuparemos cerca de 30 minutos por telefone. Portanto, considerando que um técnico ocupa 30 minutos em cada telefone, esta tarefa demoraria 3000 minutos, ou seja, 50 horas, logo a conclusão da actualização do *firmware* dos 100 telefones, seria considerada impraticável nos dias de hoje.

O Instituto Politécnico de Leiria (IPL) tem aproximadamente cerca de 500 telefones IP, assim, pretende-se que esta dissertação apresente uma proposta de gestão global para estes dispositivos, mais propriamente uma solução de provisionamento para os telefones IP. Quanto à problemática da segurança, já existe a preocupação por parte dos serviços informáticos do IPL, tanto que a infra-estrutura *ethernet* que suporta

¹*Analog Terminal Adapter* – é um adaptador de telefone analógico que permite ligar um telefone convencional a uma rede VoIP.

todos estes telefones IP já se encontra separada do resto da rede através de redes lógicas independentes denominadas de Virtual LAN (VLAN) [Anderson06]. Apesar de haver esta preocupação esta não fará parte do âmbito da presente dissertação.

1.1 O projecto VoIP@RCTS

A Fundação para a Computação Científica Nacional (FCCN) é uma instituição Portuguesa privada sem fins lucrativos que tem como principal actividade o planeamento, gestão e operação da Rede Ciência, Tecnologia e Sociedade (RCTS).

A FCCN é também a entidade competente que gere o serviço de registo do domínio de Internet .pt.

A RCTS é uma rede informática de alto desempenho que usa protocolos da Internet para garantir uma plataforma de comunicação e colaboração entre instituições do sistema de ensino, ciência, tecnologia e cultura. É utilizada também para que organismos com requisitos avançados de comunicações, efectuem testes em aplicações e serviços avançados de comunicações. A RCTS constitui no fundo uma rede de investigação e ensino nacional, que disponibiliza uma infraestrutura de comunicação para investigadores, professores e estudantes portugueses.

O projecto VoIP@RCTS tem a finalidade de equipar as instituições de ensino superior público, com ligação à RCTS, de infraestruturas necessárias ao transporte de tráfego de voz. Esta rede será totalmente segura e convergente com a estrutura de ligação à RCTS já existente. No fundo, este projecto interliga todos os sistemas telefónicos de todas as instituições aderentes, que terá como vantagem imediata, a comunicação de voz, entre elas, ser gratuita.

A FCCN estima que a implementação deste projecto conduza a uma redução de custo de cerca de 30%, tanto em chamadas telefónicas como em custo com componentes da própria infraestrutura.

1.2 Motivação e objectivos

No Instituto Politécnico de Leiria (IPL) encontra-se, à data deste documento, a decorrer a implementação de um projecto de voz para todo o Instituto. Este projecto é uma ramificação de um projecto ainda mais vasto, o VoIP@RCTS, promovido pela FCCN, projecto este já apresentado na secção 1.1.

O projecto VoIP@RCTS prevê a interligação das infraestruturas de voz, de instituições publicas aderentes, através do *backbone* de alto desempenho da RCTS, já existente nessas instituições.

No Instituto Politécnico de Leiria, o universo de telefones VoIP, poderá chegar aos 1000, visto que estão previstos cerca de 500 *hardphones* em produção, e um número ainda indefinido de *softphones*. De forma a se conseguir assegurar a manutenção de toda esta nova infraestrutura, torna-se imprescindível uma plataforma de gestão inovadora para todo o parque VoIP, caso contrário, e apesar dos custos com chamadas baixarem significativamente, subirão os custos com a manutenção.

O elevado número de terminais VoIP e a falta de um mecanismo centralizado para os administrar são a grande motivação deste trabalho. Para o efeito, propõe-se a definição e avaliação de soluções existentes no mercado, capazes de cumprir com as necessidades reais do parque VoIP do IPL. Pretende-se apresentar uma arquitectura, que num patamar a alto nível, descreva um modelo de plataforma ideal de gestão dos telefones IP presentes na instituição. Por fim, é importante planificar uma plataforma de baixo custo, se possível baseada em soluções livres, com uma interface simples de administrar.

1.3 Resumo das principais contribuições

Esta secção apresenta de forma abreviada as principais contribuições deste trabalho. Tendo em consideração os objectivos definidos na secção 1.2, essas contribuições podem ser resumidas da seguinte forma:

- Pesquisa e análise de plataformas de provisionamento já existentes, tanto livres como comerciais.
- Proposta de uma arquitectura de provisionamento de telefones IP com várias funcionalidades tais como: configurações de registo da conta, actualização de *firmware*, alteração de toque, imagem de logotipo, idioma e agenda.
- Desenvolvimento de um protótipo da arquitectura proposta, num cenário típico de laboratório.

1.4 Estrutura da Dissertação

Esta dissertação está estruturada em cinco capítulos que reflectem o trabalho desenvolvido para atingir os objectivos definidos.

No presente capítulo são apresentadas as motivações desta dissertação e definidos os principais objectivos do trabalho. São também apresentadas as principais contribuições para a área de investigação em que se enquadra este trabalho. É ainda incluída a estrutura do relatório para transmitir uma visão global da organização e conteúdo deste documento.

No capítulo 2, é efectuado um levantamento das tecnologias relacionadas com o âmbito da voz sobre IP, mais especificamente, as tecnologias ligadas ao provisionamento de terminais telefónicos IP. Após a caracterização da voz sobre IP, apresentaram-se os vários dispositivos VoIP que irão fazer parte desta dissertação, bem como algumas definições de nomenclaturas e termos usados. Posteriormente, é

definido o termo provisionamento e o seu enquadramento nesta dissertação. É apresentada a ferramenta Asterisk, o seu funcionamento e a integração que tem com interfaces *web*, para facilitar a sua gestão. É também analisada a sua interoperabilidade com equipamentos e várias tecnologias.

Ainda no capítulo 2, são apresentadas soluções de provisionamento existentes, como o *End Point Manager*, o *Voice Pulse Global Provisioning System* e também soluções comerciais. São igualmente abordadas várias técnicas utilizadas no provisionamento de telefones IP que não são mais que a utilização de protocolos tão conhecidos como: TFTP (*Trivial File Transfer Protocol*) [Sollins92], FTP (*File Transfer Protocol*) [Postel85], HTTP (*Hipertext Transfer Protocol*) [Fielding99] e DHCP (*Dynamic Host Configuration Protocol*) [Droms97]. Por fim ainda é caracterizado um protocolo de provisionamento utilizado em redes de banda larga como o ADSL (*Asymmetric Digital Subscriber Line*) [Bathrick99].

No capítulo 3 é caracterizado o funcionamento de cada modelo de telefone IP, existente no Instituto Politécnico de Leiria, que será alvo de provisionamento centralizado. No fundo constitui um estudo de como se provisiona cada um destes equipamentos.

Como resultado do conhecimento adquirido nos capítulos 2 e 3, elaborou-se uma arquitectura que será o modelo de uma plataforma de provisionamento que servirá as necessidades do IPL, apresentada no capítulo 4. De forma a realizar testes e comprovar a arquitectura proposta nesta dissertação, será apresentado, no capítulo 5, um protótipo que se desenvolveu já com algumas funcionalidades de provisionamento. Este protótipo terá também uma interface *web* que facilita o provisionamento dos dispositivos, evitando que se criem e editem manualmente ficheiros de configuração.

A conclusão de todo o trabalho realizado nesta dissertação é concretizada no capítulo 6 e por fim são apresentados alguns anexos que complementam os assuntos abordados ao longo deste documento.

2 Estado da arte

Neste capítulo serão abordadas as tecnologias que, directamente ou indirectamente, são necessárias ao desenvolvimento do trabalho, sobre o tema, da presente dissertação.

2.1 Voz sobre IP

A voz sobre IP é a unificação de duas formas de comunicação distintas (voz e dados), numa só rede convergente. Esta tecnologia, também conhecida por VoIP - do Inglês *voice over IP*, permite essencialmente, a comunicação de voz sobre uma rede de dados, baseada no protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*).

É uma tecnologia que está amplamente em grande utilização nos dias de hoje, a nível doméstico, educacional e empresarial. Esta tecnologia ganhou força como complemento à PSTN e em alguns cenários substituiu-a por completo. Esta forma de comunicação permite a utilização da mesma cablagem para ligação de computadores e telefones, facilitando tanto o projecto de toda a estrutura de rede como a manutenção da mesma. [Maximilian10]

A convergência do tráfego de voz nas comunicações de dados consegue-se normalmente à custa de DSP (*Digital Signal Processor*), que são processadores que incluem conversores analógicos/digitais. Estes conversores segmentam o sinal de voz

em quadros, armazenando-os em pacotes de dados. Esses pacotes, encontrando-se já no domínio digital, são transportados através da rede IP até ao destinatário (telefone IP ou outro dispositivo de voz), onde é consumada a operação contrária, ou seja, os pacotes de voz na forma digital são convertidos em sinais analógicos. [Silva03] [Fernandes03]

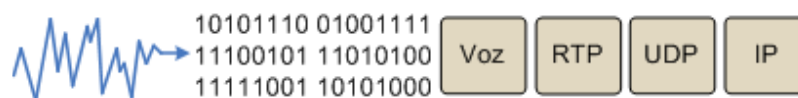


Figura 2.1 - Conversão Voz/Pacotes de dados

Para se alcançar um bom funcionamento de um sistema VoIP, é utilizado um conjunto de protocolos de controlo de sessão, que estabelecem e finalizam uma chamada. Este funcionamento não seria conseguido sem a utilização de adequados codificadores/descodificadores (*codec*) de áudio. Os *codecs* são ferramentas estabelecidas a partir de algoritmos de compressão/descompressão que permitem codificar a voz em pacotes IP ou vice-versa. Existem vários tipos de *codecs*, que variam entre si, na intensidade e na qualidade da compressão. [Silva03]

Associado à voz sobre IP estão os termos mobilidade, integração e redução de custos. Em termos de mobilidade e integração, o VoIP consegue congrega estes conceitos de uma forma mais activa que os serviços telefónicos convencionais, devido à facilidade de integração com computadores, PDA (*Personal Digital Assistants*), telefones sem fios Wi-Fi (*Wireless Fidelity*), domótica, entre outros. A redução de custos é uma certeza, por um lado, porque os operadores oferecem um plano de tarifas mais competitivo e, por outro, porque é possível tirar melhor partido das ligações de dados entre filiais, passando também comunicações de voz.

2.1.1 Compressão e codificação

Um sinal analógico é um sinal contínuo composto por variadas formas de ondas viajando pelo ar em ciclos. São exemplos de sinais analógicos a voz humana, o vídeo e a música. [Tittel96]

A voz humana é um conjunto de ondas mecânicas¹ em que as frequências principais então compreendidas na gama de 300 a 3400 Hz. Estas ondas têm alguns padrões que se repetem em função do timbre de voz e dos fonemas emitidos durante uma conversação. [Cardoso06]

Num ambiente totalmente analógico é possível assegurar uma boa qualidade, uma vez que, a transmissão das ondas entre os interlocutores são efectuadas através de um meio metálico, onde é possível usar amplificações analógicas. Contudo, esta forma de transmissão representa um custo muito elevado devido à impossibilidade de se utilizar o mesmo meio para transmitir mais de um canal, em simultâneo. [Solari97]

Para transmitir áudio sobre uma rede informática é necessário converter as formas de onda analógicas, que compõem a voz, em formato digital. Para tal, a voz passa pelos tradicionais processos de conversão analógica-digital que são conhecidos por: amostragem, quantificação e codificação. Só depois de se ter a voz em formato digital é que se podem inserir estes pacotes no campo de dados do protocolo de transporte. Após os pacotes de dados/voz serem transportados na rede e chegarem ao seu destino, é efectuada a conversão inversa, ou seja, conversão digital-analógica, para deste modo se tornar audível ao ouvido humano.[Silva03]

O objectivo de qualquer método de codificação de voz é produzir um conjunto de códigos binários que necessitem de uma taxa, o mais baixa possível de transmissão, de modo a que o destinatário possa reconstituir o sinal de voz, mais próximo possível do original, assegurando assim uma boa qualidade.[Solari97]

A diversidade de *codecs* existente tem a ver com o método de codificação utilizado. A tabela 2.1 apresenta alguns dos *codecs* mais utilizados, onde se pode ver a largura

¹Ondas mecânicas – são ondas produzidas por uma vibração num material, como por exemplo, uma onda na água, a vibração de uma corda de viola ou até mesmo a voz de uma pessoa.

de banda ocupada (kbps), o atraso do algoritmo de compressão de cada um e o *Mean Opinion Score* (MOS).[Sahabudin09][Minoli98][Rodman08]

Codec	Taxa de bits Unidades (kbps)	Latência (ms)	MOS
G.711	64	0,125	4,1
G.726	32	0,125	3,85
G.729	8	15	3,7
G.723.1	5,3 / 6,3	37,5	3,6

Tabela 2.1 - Codecs [Sahabudin09][Minoli98][Rodman08][Macian08][Gibson05]

O MOS define uma escala de 0 (mau) a 5 (excelente) com o objectivo de medir a qualidade da voz. Este valor é obtido fazendo a média de avaliações efectuadas por diversas pessoas sujeitas ao teste de qualidade. [Gonia10]

A latência apresentada na tabela 2.1 é um valor fixo visto ser um parâmetro inerente ao algoritmo de compressão do próprio *codec*. Este atraso na propagação do pacote de voz é normalmente superior devido à introdução de latência do tipo de comunicação usada (*Ethernet*, *WiFi*, *ATM*, *Frame Relay*, etc).

Para uma boa qualidade e percepção da comunicação entre duas pessoas, o atraso total (ponto-a-ponto) não deverá ser superior a 150ms. Entre 150ms e 200ms, pessoas mais sensíveis já poderão notar o atraso. A partir de 250ms já se tornará um atraso bem perceptivo e a comunicação começará a fazer-se com deficiências. Um caso bem prático é os intervenientes começarem a sobrepor as suas falas um ao outro.[Gonia10]

Estes valores de Taxa de Bits, latência e MOS podem auxiliar no dimensionamento e implementação de sistemas de voz sobre IP onde a largura de banda é limitada. Pode mesmo auxiliar a resolução de problemas, ajudando a parametrizar os dispositivos de voz em certos segmentos da rede.[Gonia10]

É essencial nos dias de hoje, quando se implementam sistemas de voz sobre IP, assegurar a Qualidade de Serviço (QoS) na estrutura da rede. A QoS permite, muito

sucintamente, que se possa diferenciar tráfego numa rede de IP, dando por exemplo prioridade a pacotes do tipo VoIP face a outros dados.[Gonia10]

2.1.2 Protocolos de sinalização

Conforme abordado no início deste capítulo, as chamadas de VoIP, usufruem da rede IP para a transmissão de sinais de Voz, em tempo real, na forma de pacotes de dados. Para isto ser possível, foi necessário desenvolver protocolos adicionais de sinalização de chamadas e transporte de voz que permitissem a comunicação com qualidade aproximada à fornecida pelas redes de voz convencionais. São exemplo destes protocolos, o H.323, o SIP e o IAX, que serão abordados mais à frente.

Nesta secção, irão abordar-se os protocolos de sinalização mais importantes na evolução do VoIP: H.323, SIP e IAX. Esta abordagem será algo superficial uma vez que não é objectivo desta dissertação conhecer os protocolos ao seu mais baixo nível.

2.1.2.1 H.323

O H.323 é um padrão que rege uma série de sub-sistemas, protocolos, e algoritmos para comunicações multimédia. Originalmente, este padrão surgiu aproveitando normas já existentes (H.320¹ e H.324²) para comunicações de vídeo através de redes de comutação de circuito, como por exemplo a linha RDIS (Rede Digital com Integração de Serviços).

É uma recomendação que integra um grupo de recomendações H.32x definida pela *International Telecommunication Union Telecommunication Standardization* (ITU-T), equipa responsável pelo tratamento de Sistemas Audiovisuais e Multimédia. A recomendação H.323 tem a finalidade de especificar as comunicações multimédia nas redes baseadas em pacotes que não sejam dotadas de Qualidade de Serviço(QoS). Especifica também, padrões de codificação e decodificação dos pacotes de dados

¹H.320 – é uma recomendação do ITU-T para transmissão de dados multimédia sobre linhas RDIS.

²H.324 – é uma recomendação do ITU-T para transmissão de dados multimédia sobre linhas analógicas.

áudio e vídeo, garantindo desta forma que produtos baseados no padrão H.323 de um fabricante comuniquem com produtos H.323 diferentes.[ITU10]

O H.323, foi criado tendo como linhas orientadoras o áudio/videoconferência e as capacidades de processamento de dados, numa rede de comutação de pacotes.

De forma mais específica, o H.323 define um conjunto de protocolos para comunicações de áudio e vídeo composto pelos seguintes protocolos:

- H.245 para controlo;
- H.225.0 para a ligação e estabelecimento da comunicação;
- H.332 para largas conferências;
- H.450.x para todos os serviços adicionais disponibilizados pelo H.323.

Para um sistema H.323 funcionar na íntegra são necessários alguns elementos de rede, os quais são descritos na tabela 2.2.

Elemento	Função
Terminal	São os dispositivos de fim de linha (<i>Endpoints</i>) que providenciam comunicação bidireccional em tempo real com outro terminal H.323.
Gateway	Normalmente um Gateway é composto por um Media Gateway Controller (MGC) e um Media Gateway (MG). Um MGC trata da parte de sinalização da chamada e o MG trata de toda a parte de média (vídeo e áudio). Os Gateways também podem fazer a interligação com outras redes (PSTN, Servidores H.323, Proxy).
Gatekeeper	O Gatekeeper fornece serviços de resolução de endereços e controlo de acesso à rede para terminais H.323, MCU e Gateways. Tem a capacidade de colocar chamadas directamente entre dois terminais, ou encaminha-la para si mesmo e aplicar funções como ocupado, correio de voz, etc...
<i>Multipoint Control Units (MCU)</i>	São responsáveis por gerir conferências multi-ponto entre dois ou mais terminais.

Tabela 2.2 - Função dos elementos de rede do H.323

2.1.2.2 SIP - Session Initiation Protocol

O SIP é um protocolo de sinalização, presença¹ e mensagens instantâneas² desenvolvido para estabelecer, modificar e terminar sessões multimédia. Este protocolo acaba por usufruir da vantagem da arquitectura de Internet e do conjunto de protocolos que a compõem, como é o caso do UDP (*User Datagram Protocol*), TCP (*Transmission Control Protocol*), IP, DNS (*Domain Name System*), entre outros. [Johnston04] A sua estrutura, como seria de esperar, é baseada em elementos do *Hyper Text Transport Protocol* – HTTP [Fielding99] e, a sua codificação, efectuada em texto. [Handley99]

Foi desenvolvido com o objectivo focado na comunicação entre dispositivos multimédia em que a comunicação pode ser uma sessão interactiva entre dois ou mais intervenientes, podendo envolver intercâmbio de vídeo, voz, mensagens instantâneas, jogos e realidade virtual.

O SIP é caracterizado pela sua principal função de iniciação de sessão (*session initiation / setup*) baseado no modelo cliente-servidor, onde os intervenientes durante uma comunicação normalmente trocam de posições (cliente ↔ servidor). [Sinnreich01]

Este protocolo foi criado pelo grupo de trabalho *Multiparty Multimedia Session Control* (MMUSIC) do *Internet Engineering Task Force* (IETF³). A primeira versão deste protocolo foi publicada em 1997 sob a forma de *Internet-Draft*⁴.

Depois de algumas alterações significativas no protocolo, surgiu a segunda versão, publicada como *Internet-Draft*, em 1998. Em Abril de 1999, o SIP foi tornado num padrão pela RFC 2543. Foram publicadas posteriormente correcções que deram origem a outras RFC, como é o caso da obsoleta RFC 3261. [Johnston04] [Handley99][Rosenberg02]

¹Presença – refere-se à troca de informação em tempo real sobre o estado de um utilizador/contacto.

²Mensagens instantâneas – refere-se à trocas de mensagens, normalmente de texto em tempo real entre dois utilizadores.

³<http://www.ietf.org>

⁴*Internet-Draft* - São documentos publicados pelo IETF, considerados esboços, que poderão dar origem a RFC (*Request For Comments*).

O protocolo SIP é essencialmente constituído pelos seguintes elementos: SIP *User Agent*, SIP *Server*, SIP *Redirect Server*, SIP *Registrar Server* e SIP *Location Server*. Estes elementos, em conjunto, permitem a entrega das mensagens entre os intervenientes da comunicação e são brevemente descritos na tabela 2.3[Handley99].

Elemento SIP	Função
User Agent	São os dispositivos terminais, tais como, telefones móveis, auriculares (<i>handsets</i>), Pcs, PDAs, etc. São estes que estabelecem e gerem a sessão SIP.
Proxy Server	Aceita pedidos do <i>User Agent</i> (UA), e solicita ao SIP <i>Registrar Server</i> informação sobre o UA destinatário. O <i>Proxy Server</i> , encaminha o convite da sessão para o destinatário se estiver no mesmo domínio, caso contrário, encaminha para o <i>Proxy Server</i> do destinatário.
Redirect Server	Possibilita aos SIP <i>Proxy Servers</i> direccionarem convites de sessões SIP para domínios externos.
Registrar Server	Contém as base de dados com a localização dos utilizadores de um determinado domínio.

Tabela 2.3 - Funções dos elementos SIP

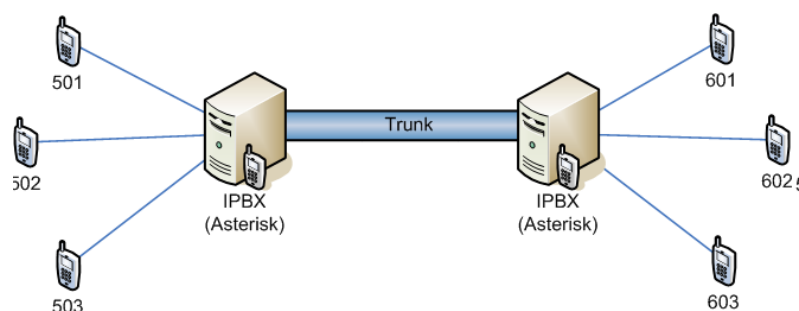
2.1.2.3 IAX - Inter Asterisk Exchange

Inicialmente, o protocolo IAX foi desenvolvido pela Digium¹ para criar ligações entre servidores Asterisk. Estas ligações denominadas por *trunks* interligam dois PBX Asterisk de forma a encaminharem chamadas entre si, ou seja, pacotes de sinalização e tramas² de voz. Na figura 2.2 é possível ver um exemplo de uma ligação em *trunk*. Na figura estão representados dois planos de numeração, um que se inicia por cinco (5xx³) e outro por seis (6xx). Com estes dois planos diferentes consegue-se criar uma regra em cada Asterisk, que indique que todas as chamadas começadas por cinco (5xx) são encaminhadas pelo *trunk* e o mesmo acontece para o plano de numeração iniciado por seis (6xx).

¹[Http://www.digium.com](http://www.digium.com)

²Trama – é uma porção de informação transmitida através de uma rede de dados. O mesmo que quadro de dados, *stream* ou *frame*.

³5xx – no Asterisk indica todos os números marcados que se iniciam por 5, sendo os dois seguintes constituídos por quaisquer outros algarismos.

Figura 2.2 - Exemplo de uma ligação *trunk*

O IAX é um protocolo baseado no SIP, logo, existe uma certa semelhança entre ambos. É o protocolo nativo do Asterisk que para além de *trunks* suporta também transporte de sinalização e dados em UDP. Em cenários com um IPPBX Asterisk e terminais IP clientes com suporte para IAX, é possível substituir por completo os protocolos SIP e H.323.[Spencer10]

Como actualmente o IAX se encontra na versão 2 com a RFC 5456, qualquer referência neste documento dirá respeito a essa mesma versão.

Os principais objectivos que levaram à criação deste protocolo foi a redução de largura de banda entre os dispositivos de comunicação VoIP, evitar problemas de *Network Address Translator* (NAT) e suportar a troca de informação de planos de extensões (*dialplans*).

A redução de largura de banda por parte do IAX foi conseguida usando um protocolo binário, ao invés de mensagens de texto, como acontece no SIP.

De forma a evitar problemas de NAT, o IAX faz uso do protocolo de transporte UDP, tipicamente na porta 4569¹, para transporte de sinalização e dados, ou seja, é utilizado o mesmo protocolo na mesma porta para as duas situações. Desta forma, o IAX consegue, com menos probabilidade de falhas, passar por encaminhadores (*Routers*) e *firewalls*, face ao SIP. [Abbassi06] [Chava08]

¹A primeira versão do IAX, utilizava o porto 5036

2.2 Dispositivos e equipamentos VoIP

Neste sub-capítulo, irão abordar-se alguns termos técnicos utilizados em equipamentos físicos de voz sobre IP (VoIP), que poderão encontrar-se nos campi do Instituto Politécnico de Leiria.

2.2.1 Telefones IP (*Hardphones*)

Telefones IP são dispositivos que permitem efectuar comunicações de voz ou vídeo sobre uma rede IP (VoIP). O termo “telefone IP” pode ser um pouco mais abrangente e referir-se também a telefones virtuais (ou *softphones*), no entanto, neste documento, este termo referir-se-á a um dispositivo telefónico físico.

Também é correcto e comum chamar terminal IP, a estes equipamentos, sendo neste caso um termo mais genérico pois pode englobar outros dispositivos, que não sejam telefones, como por exemplo um adaptador analógico-digital (ATA).

São diversos os fabricantes deste género de equipamentos. No IPL existem quatro modelos de telefones IP que serão contemplados no modelo de provisionamento a desenvolver. Estes telefones IP estão apresentados na tabela 2.4, onde estão especificadas as unidades existentes no IPL de cada um dos modelos.

Marca	Modelo	Unidades
Polycom	Soundpoint IP 330	320
Snom	360	50
Linksys	SPA942	100
Grandstream	BT200	20

Tabela 2.4 - Telefones IP existente no IPL

No parque VoIP do IPL fazem também parte outros equipamentos VoIP, como por exemplo: interfaces ATA, IPPBX e Médias Gateway¹.

¹*Media Gateway* – é normalmente um equipamento ou ferramenta que faz a interligação entre dois sistemas de comunicação, por exemplo, entre uma rede VoIP e a Rede Pública de Telefonia Comutada (PSTN).

Nas secções que se seguem, será apresentado o estudo aprofundado dos telefones IP da instituição. Este estudo teve a finalidade de conhecer as características dos terminais bem como a forma de os provisionar remotamente. Portanto, as soluções aqui apresentadas, são vocacionadas para provisionamento remoto, logo, o relatório não inclui outras formas de configuração em que o resultado seja idêntico, por exemplo, configuração manual no telefone ou configuração via servidor *web* do terminal IP.

Existem diversos telefones IP disponíveis no mercado com capacidade de comunicarem através dos protocolos H.323, SIP, IAX ou outros, podendo incorporar diversas funcionalidades e características, tais como:

- Vídeo;
- Registo de várias linhas/extensões;
- *Codecs*;
- Toques polifónicos personalizáveis;
- Agenda (local e sincronizada com AD¹);
- Conferência com três ou mais intervenientes;
- Mãos livres;
- Teclas programáveis;
- Configuração via *web* e XML;
- Provisionamento centralizado;

Na figura 2.3 é apresentado um exemplo de um telefone IP onde se pode confirmar, as suas características mais evoluídas face a um telefone analógico convencional, como é o caso do visor de grandes dimensões, as várias teclas de atalho e de acesso ao menu de configuração.

¹*Active Directory* – Serviço de directório para autenticação centralizada da Microsoft.



Figura 2.3 - Exemplo de um *hardphone*

2.2.2 Telefones virtuais (*Softphones*)

Um telefone virtual ou *softphone* é uma aplicação informática que corre num PC com o objectivo de conjuntamente com uns auscultadores e microfone simular um telefone físico. Uma grande vantagem de um *softphone* é a portabilidade, uma vez que basta ter a aplicação em qualquer computador em qualquer local com acesso à rede, efectuar as configurações necessárias, iniciar sessão com as credenciais próprias e a extensão estará disponível. Outra vantagem é o custo, visto haverem diversos de utilização livre. Desta forma, com um computador portátil e com uma ligação à Internet é possível prolongar uma determinada extensão a qualquer localização, em qualquer país do mundo.[Jason05]

Na figura 2.4, encontra-se representado um exemplo de um *softphone*, o X-lite¹, um dos mais conhecidos pelos utilizadores de sistemas VoIP.

¹[Http://www.counterpath.com](http://www.counterpath.com)



Figura 2.4 - Telefone virtual IP X-lite

O X-lite é um *softphone* do fabricante CounterPath e apesar de ser livre apenas para utilizadores domésticos, é importante apresentá-lo neste documento por duas razões. Uma delas deve-se ao facto de ser um dos mais utilizados por utilizadores de voz sobre Internet a nível mundial.[Farrell09]

A outra razão tem a ver com o projecto da FCCN, o ARARA¹, que o IPL está a aderir, em que o *softphone* fornecido será deste fabricante.

O projecto ARARA tem como funcionalidades permitir através de um único terminal aplicacional disponibilizar voz, vídeo-chamada, audioconferência, videoconferência e mensagens instantâneas sobre protocolos universais como o SIP e o XMPP (*Extensible Messaging and Presence Protocol*)²[Andre04].

Outro exemplo de um *softphone* está apresentado na figura 2.5 - Ekiga³. Este *softphone* é distribuído sob uma Licença Pública de Utilização (GPL) e tem versões tanto para Windows como para Linux. As suas funcionalidades são semelhantes ao X-lite.[Yang09]

¹Projecto ARARA, www.fccn.pt/arara

²XMPP, é uma arquitectura livre, segura e descentralizada para comunicação em tempo real sobre a Internet.

³[Http://ekiga.org](http://ekiga.org)

A vantagem que mais se destaca no Ekiga em relação ao X-lite é realmente a sua licença - GPL, podendo ser utilizado para fins domésticos e comerciais ao contrário do X-lite que é livre apenas para utilizadores domésticos.[EK10]



Figura 2.5 - *Softphone* ou Telefone virtual IP Ekiga

Um *softphone* pode ter todas as funcionalidades de um telefone convencional, nomeadamente: encaminhamento de chamadas, chamada em espera, marcação rápida e tele-conferência. Tipicamente, os *softphones* estão dotados de muitas outras funcionalidades como: vídeo, várias linhas e protocolos de comunicação, compressão e codificação, mensagens instantâneas e alguns até permitem envio de SMS (*Short Message Service*).[EK10]

Estas aplicações VoIP seguem o modelo de arquitectura Cliente-Servidor para a parte de autenticação e alguns processos secundários. Já os pacotes de voz são encaminhados directamente para o destinatário, ou seja, usando a arquitectura *peer-to-peer*. Sendo assim, o funcionamento desta aplicação é semelhante a um telefone

IP, precisando de se registrar num servidor (IPPBX Asterisk ou outro) para conseguir comunicar com outro dispositivo. Existem diversas empresas na Internet que disponibilizam serviços para comunicação onde é possível utilizar *softphones* proprietários. De qualquer forma, se estes serviços utilizarem protocolos padrão é possível usar *softphones* como o X-lite e o Ekiga. Alguns destes serviços até oferecem chamadas gratuitas, normalmente, com uma duração de tempo limitado. [Robert08]

2.2.3 Adaptador ATA

Um adaptador ATA é um dispositivo que permite utilizar um telefone convencional (POTS – *Plain Old Telephone Service*) numa rede de voz sobre IP. O ATA faz a conversão dos sinais de voz analógicos para o domínio digital, enviando-os para o destinatário, através da rede IP.

Na sua essência, o ATA incorpora as funcionalidades de um telefone IP numa pequena caixa onde se liga o telefone analógico e a ligação Ethernet (ou outra).

As ligações ao meio de comunicação mais usadas são via interface Ethernet, USB ou xDSL. Na figura 2.6 é possível visualizar as ligações normalmente disponíveis num adaptador ATA.

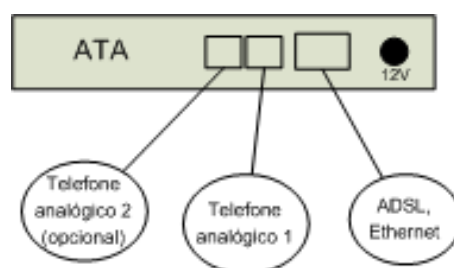


Figura 2.6 - Ligações disponíveis num adaptador ATA

Este conceito de dispositivo ATA pode vir incorporado em equipamentos como *routers*, *modems*, entre outros.

2.2.4 Telefones de conferência

Os telefones de conferência, que normalmente reúnem as mesmas funcionalidades que os telefones IP, são dispositivos mais direccionados para serem utilizados em salas de reuniões pois possibilitam a vários intervenientes comunicarem em áudio-conferência. Tipicamente, estes telefones prescindem de auscultadores, estão equipados com mais que um microfone e funcionam em alta-voz. As restantes características, em termos de interfaces de comunicação, configuração e manuseamento assemelham-se aos telefones IP típicos.

Na figura 2.7 é apresentado um telefone de conferência da marca Polycom, cujo formato foi desenhado para estar no centro de várias pessoas. É caracterizado pelo seu formato em triângulo que ajuda a direccionar os microfones para os intervenientes.



Figura 2.7 - Telefone de conferência Polycom

2.3 Provisionamento

Provisionamento define um processo de preparação, configuração e/ou personalização de um ou mais dispositivos, de forma centralizada, através de um servidor, via navegador *web* ou mesmo via *scripts*. O termo provisionamento pode referir-se a configurações efectuadas no servidor *web* do equipamento, sendo desta forma feitas uma a uma.

No âmbito desta dissertação, interessa o provisionamento centralizado e automático, ou seja, aquele em que as configurações ou personalizações podem ser feitas em massa a todos os equipamentos, através de um ponto central - servidor. Para este último caso, é usualmente utilizada a designação auto-provisionamento em várias webgrafias¹.

Na área das telecomunicações é comum definir provisionamento quando se disponibilizam dispositivos, serviços ou mesmo produtos ao utilizador final, sem que este tenha que intervir, uma vez que tudo é feito remotamente.

Também é designado de provisionamento o processo que providência, aos utilizadores e dispositivos, o acesso a dados ou a tecnologias a eles inerentes. Pode dar-se o exemplo de um utilizador que ingressa numa empresa, e a sua primeira tarefa ser a leitura das regras e manuais de procedimentos de acesso à rede, sem que para isso seja necessário a ajuda do administrador de sistema. No fundo, este caso exemplifica uma forma de provisionamento.

É na automação da configuração de terminais VoIP que se vai focalizar esta dissertação, ao ponto do cliente final só precisar de ligar o telefone IP, não conhecendo nada sobre a sua parametrização, e este ficar totalmente operacional. É de realçar que, um sistema de provisionamento faz mais sentido quando o parque de terminais é em grande número. Imagine-se o trabalho e tempo ganho numa organização com uma grande quantidade de telefones. Haveria muito trabalho que seria necessário despendar para configurar e actualizar o *firmware* ou alterar e personalizar todos os telefones, manualmente, um a um.

De seguida, para além da abordagem introdutória ao Asterisk, serão também apresentadas algumas soluções de provisionamento disponíveis no mercado.

¹Webgrafia, refere-se a pesquisas na Internet.

2.3.1 Asterisk

Um PBX é o equipamento que gere os telefones e toda a comunicação de voz de uma determinada instituição. Normalmente é conhecido por central telefónica. Um PBX tradicional é geralmente constituído por uma componente hardware, com cartas de extensões e, uma componente de software ou sistema operativo, onde se efectua toda a configuração do plano de numeração, entre outras opções. Este modelo de central telefónica, embora bastante funcional e estável, sempre foi um pouco fechado a expansões e adição de novas tecnologias pois, cada fabricante tinha o seu sistema próprio que diferia de todos os outros. Apesar de hoje em dia estes PBX já possuírem, na sua grande maioria, cartas de expansão VoIP, os PBX baseados em rede IP (IPPBX) estão a ter cada vez maior aceitação de mercado actual. Na opinião do autor da presente dissertação, que não é mais que uma reflexão que resultou das mais diversificadas pesquisas, o sucesso dos IPPBX deveu-se muito à própria evolução do Asterisk e também do protocolo SIP.

O Asterisk é um IPPBX de código livre baseado em Linux e tem, entre outras funções, a possibilidade de gerir chamadas telefónicas de voz sobre uma rede IP, interoperabilidade com outros sistemas (PBX tradicionais), suporte para vários protocolos (H.323, SIP, IAX) e ainda disponibilização de diversos serviços, tais como: correio de voz, correio de voz integrado com correio electrónico, ACD¹(*Automatic Call Distribution*), IVR²(*Interactive Voice Response*), chamadas em conferência, música em espera, entre outras. [Rodrigues06]

Esta ferramenta foi criada por Mark Spencer, em 1999 e, nos dias de hoje, é a grande base da maioria das distribuições livres de um IPPBX. Pode-se afirmar que o Asterisk foi o grande impulsionador dos IPPBX devido a ser uma ferramenta gratuita e com a possibilidade de inúmeros serviços e integração com PBX convencionais através de placas fabricadas pela Digium. [Wong06]

¹É a possibilidade de encaminhar chamadas para um grupo consoante determinadas regras.

²É um menu de voz interactivo que possibilita a um interlocutor seleccionar uma determinada opção (departamento, serviço, falar com a operadora, etc).

A sua distribuição é feita sob GPL – Licença de Público Geral, uma das razões que ajudou a impulsionar a sua difusão. Outro facto, para a sua difusão em massa no mundo das comunicações de voz, também se deve ao surgimento de múltiplas distribuições baseadas em Asterisk com interfaces *web* simples de utilizar, tais como: TrixBox¹, Elastix², FreePBX³, AsteriskNow⁴, IPbrick⁵, Edgebox⁶, entre outros.

A constituição do Asterisk é composta por um núcleo que trata essencialmente das funções de comutação, codificação e decodificação, execução de comandos, gestão e escalonamento de entrada e saída. Num nível mais alto encontram-se as API do Asterisk que permitem a interligação com o núcleo e tornar mais acessível a programação de novas funcionalidades por programadores.[Schwarz04]

As interfaces *web* que complementam o Asterisk e facilitam grandemente a sua gestão, configuração e parametrização, permitem diversas funcionalidades. Por esta razão, deixa de ser necessário a configuração do IPPBX via linha de comandos e edição de ficheiros de texto, recorrendo-se apenas a uma consola *web*. A interface *web* simplifica de tal forma o Asterisk que praticamente qualquer técnico de TI (Tecnologias de Informação), hoje em dia, instala e configura um sistema de voz sobre IP baseado nesta ferramenta.

Na tabela 2.5 é possível conhecer algumas das funções que as interfaces *web* podem desempenhar.

¹<http://www.trixbox.org>

²<http://www.elastix.org>

³<http://www.freepbx.org>

⁴<http://www.asterisknow.org>

⁵<http://www.ipbrick.com>

⁶<http://www.edgebox.net>

Directório/pasta	Função
Extensões / <i>Extensions</i>	Criação e gestão das linhas, extensões e correio de voz
Troncos / <i>Trunks</i>	Manutenção de ligações entre servidores Asterisk ou outros sistemas, denominados por troncos
Rotas de saída / <i>Outbound routes</i>	Gestão das rotas (troncos) a utilizar para estabelecimento das chamadas
Rotas de entrada / <i>Inbound Routes</i>	Verifica o destino das chamadas que chegam dos troncos
Rotas DDI ¹ / <i>DID routes</i>	Identificação de uma rota de destino baseado no número marcado
Filas / <i>Queues</i>	Gestão das chamadas que estão em fila ou em espera
Música em espera / <i>Music on hold</i>	Configuração no IPPBX da música em espera
Cópias de Segurança / <i>Backups</i>	Permite a realização de cópias de segurança das configurações do sistema, bem como, reposição das mesmas
Configurações gerais / <i>General Settings</i>	Actualizações do sistema, configurações de rede, reiniciar o Asterisk, entre outras

Tabela 2.5 - Algumas funcionalidades de uma interface *web* do Asterisk

A figura 2.8 representa a interoperabilidade do Asterisk com terminais de várias tecnologias (SIP, H.323, IAX, PC, POTS²), com terminais na Internet e mesmo com outro Asterisk, tudo assente numa rede de IP.

Os “POTS *phone*” representam os telefones analógicos convencionais, que para comunicarem com o Asterisk é imprescindível algo que os interligue (PBX Gateway ou uma ligação POTS). O PBX Gateway pode ser uma ligação Ethernet da própria central telefónica (PBX) ou uma placa no servidor Asterisk com portas POTS ligadas a uma extensão do PBX.

¹DDI ou DID – *Direct Dial-In* ou *Direct Inward Dialing*, é uma funcionalidade disponibilizada pelos operadores de telecomunicações que permite a ligação de fora directamente para uma extensão, caso esta tenha um DDI atribuído pelo PBX.

²*Plain Old Telephone Service* – Representa as linhas convencionais de telefones (analógico, RDIS), e por vezes refere-se à rede pública telefónica – PSTN.

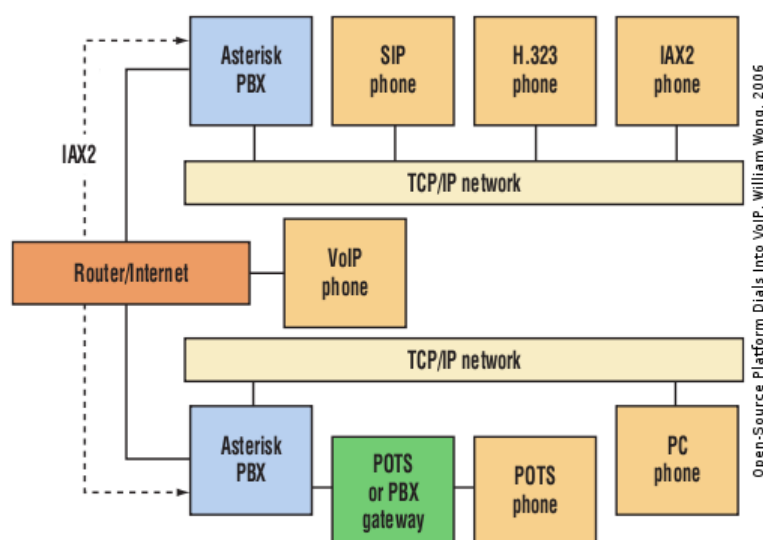


Figura 2.8 - Interoperabilidade do Asterisk

2.3.2 FreePBX

Conforme já abordado, o FreePBX é uma interface gráfica em formato *web* onde se efectua grande parte da gestão do Asterisk.

A figura 2.9 mostra a relação ou o posicionamento do Freepbx perante o Trixbox, o Asterisk e o sistema operativo (Linux). Por vezes, gera-se confusão entre o Freepbx e o Trixbox, assim, este diagrama serve para elucidar as diferenças entre ambos.

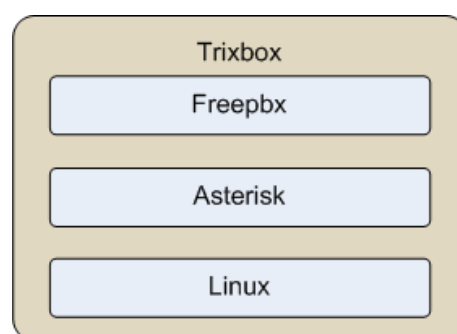


Figura 2.9 - Diferença entre Trixbox e Freepbx

Analisando a figura 2.9 verifica-se que o sistema operativo comporta o Asterisk e o Freepbx. Pode-se afirmar que o Asterisk é núcleo ou o motor de todo o IPPBX. O

Freepbx está na camada do utilizador, pois permite-lhe efectuar as configurações no Asterisk sem que seja necessário ter um conhecimento profundo sobre o mesmo. O Trixbox é o conjunto de todas as ferramentas que compõem este IPPBX que, para além do Asterisk e do Freepbx, também inclui o próprio sistema operativo - CentOS¹, o motor de base de dados – MySQL², um servidor *web* – Apache, uma plataforma servidora para interpretar as páginas *web*, programadas em PHP³ e, por último, várias bibliotecas e *scripts* que também são necessários a toda a solução.

O Freepbx disponibiliza um sistema de provisionamento de telefones na sua consola denominado por *End Point Manager*. Este aplicativo do Freepbx suporta as marcas de telefones IP Snom, Polycom, Aastra, Linksys e Granstream. De qualquer não se encontrou informação concreta quanto a modelos específicos.

O modo de funcionamento desta ferramenta é baseada na construção de ficheiros XML, que pode ser efectuado quando se adiciona o telefone à base de dados de dispositivos do Trixbox. A adição de dispositivos pode-se resumir em três formas distintas:

- Manual, escolhendo o fabricante e modelo, e inserindo o endereço MAC;
- Usando um leitor de código de barras, para ler o endereço MAC e o modelo do telefone;
- Varrendo a rede à procura de dispositivos VoIP.

Qualquer um destes métodos cria os ficheiros de configuração baseado no endereço MAC do dispositivo, guardando-os de seguida no directório do serviço TFTP.

¹[Http://www.centos.org](http://www.centos.org)

²[Http://www.mysql.com](http://www.mysql.com)

³[Http://www.php.net](http://www.php.net)

2.3.3 Voice Pulse Global Provisioning System (VGPS)

O *Voice Pulse Global Provisioning System* é uma plataforma *web* de provisionamento baseada na concepção de ficheiros de configuração específicos para um determinado fabricante do dispositivo VoIP. É uma ferramenta construída em linguagens de programação livres, tais como: PHP, MySQL e Perl¹.

Não se deu grande importância a esta solução pois segundo o *sítio web* oficial² deste projecto, não existe qualquer actualização efectuada, desde 2006. Desde essa data que a fonte do projecto indica que este está em fase experimental/desenvolvimento. Por esta razão e também devido à escassez de documentação e implementações comprovadas, esta solução foi consequentemente posta de parte.

2.3.4 Soluções comerciais

Uma das tarefas desta dissertação passava por estudar várias opções de provisionamento que pudessem servir as necessidades do IPL, inclusive soluções comerciais. Esta tarefa revelou-se um pouco complexa de executar devido à falta de informação e de opções de aplicações completas, isto é, que suportassem todos os modelos de terminais IP existentes no IPL.

Para além da pesquisa na Internet, contactaram-se também alguns fornecedores que eventualmente pudessem ter soluções de provisionamento para telefones VoIP.

Na tabela 2.6 encontra-se um resumo de soluções comerciais de provisionamento bem como os modelos de telefones IP suportados. Estas soluções aqui apresentadas fazem parte de soluções completas de telefonia IP, não sendo unicamente de provisionamento, ou seja, no fundo são apenas um módulo de toda a solução.

¹<http://www.perl.org>.

²<http://vgps.sourceforge.net>

Nome da Solução	Fabricante	Telefones IP suportados
Cisco Unified Provisioning Manager	Cisco	Cisco, Linksys
Edgebox	Critical Links	Polycom, Aastra, Grandstream, Linksys
Trixbox Pro – Hands-Free Auto-Provisioning	Fonality	Aastra, Polycom
VoiceIP-ACS	AVSystem	Só suporta terminais IP com norma TR-069
VigorACS	Draytek	Só suporta terminais IP com norma TR-069

Tabela 2.6 - Soluções comerciais com módulo de provisionamento

Analisando a tabela 2.6 verifica-se que nenhuma das opções suporta todos os modelos existentes no IPL, sendo que a aplicação da Critical Links é a que mais se aproxima, faltando apenas suportar a marca Snom. A ferramenta de provisionamento da Cisco, como era de esperar, apenas suporta os equipamentos da sua própria marca. [CiscoUPM09]

A solução *Hands-Free* da Fonality¹, a empresa detentora do Trixbox², que nos primeiros estudos parecia promissora, até à data da escrita desta dissertação só suportava, na íntegra, duas marcas de terminais IP.[TB10]

As duas soluções restantes são vocacionadas para equipamentos que incorporam exclusivamente a norma TR-069³, a qual não é suportada pelos equipamentos existentes no IPL.

2.4 Técnicas de provisionamento

Nesta secção será apresentada a investigação efectuada a técnicas empregadas no provisionamento de terminais VoIP, onde se irá expor o método usado na preparação

¹[Http://www.fonality.com](http://www.fonality.com)

²[Http://www.trixbox.org](http://www.trixbox.org)

³TR-069 – é uma norma de provisionamento sobre linhas ADSL. Esta norma será abordada na secção 2.5.

do dispositivo para ser provisionado e os métodos de transferência de ficheiros de configuração e actualizações de *firmwares*.

A grande maioria dos terminais VoIP são personalizados através de ficheiros de configuração, normalmente, em formato XML. Cada modelo tem uma sintaxe de código XML próprio, dificultando desta forma a uniformização desta área da voz sobre IP, e obrigando, ao estudo individual de cada equipamento.

O funcionamento normal de um sistema de provisionamento baseia-se na disponibilização dos ficheiros de configuração, previamente preparados. Aos terminais IP quando estão na fase de arranque é-lhes atribuídas, pelo serviço DHCP, as configurações de rede onde se incluem parâmetros que indicam onde estão os ficheiros de provisionamento.

2.4.1 TFTP

O *Trivial File Transfer Protocol* ou TFTP é um protocolo cuja finalidade é permitir transferência de ficheiros e, normalmente é muito utilizado para descarregar *firmwares* para encaminhadores, comutadores, terminais IP, entre outros. É um protocolo baseado no FTP, mas com um funcionamento ainda mais simples, pois não tem autenticação nem mecanismos de encriptação. [Sollins92]

Este protocolo também é habitualmente usado para permitir a equipamentos que não tenham sistema operativo, arrancarem via rede, utilizando para isso, a funcionalidade PXE¹(*Preboot eXecution Environment*) [Johnston06] das placas de rede e o sistema operativo no servidor TFTP.

O protocolo de transporte que o TFTP utiliza é o UDP, logo, como não é estabelecida uma comunicação, não são garantidas entregas de pacotes de dados, relegando esse tratamento para o cliente e servidor, de forma a que os dados cheguem íntegros ao seu destino. O porto UDP que o protocolo TFTP utiliza é o 69, sendo necessário

¹é um mecanismo que juntamente com o DHCP e o TFTP permitem que por exemplo um PC arranque sem sistema operativo.

parametrizar uma excepção na *firewall* caso esta exista entre o servidor TFTP e os clientes (terminais IP).

Devido às suas vantagens, nomeadamente a simplicidade e a rapidez, este protocolo é o mais utilizado no provisionamento de terminais VoIP, para o método de transferência de ficheiros (*firmware*, configuração, toques, imagens).

As configurações de rede, bem como do servidor TFTP, são atribuídas ao telefone IP pelo servidor DHCP quando este inicia. Depois do arranque inicial, já com as configurações de rede, o terminal IP vem normalmente preparado de fábrica para descarregar o *firmware*, no caso de existir uma versão mais recente, e os ficheiros de configuração. Este processo permite, portanto, provisionamento em massa de terminais IP, pois todos carregam o mesmo *firmware* e as mesmas configurações genéricas que constam no servidor TFTP. Para configurações que são únicas de um telefone, normalmente, o telefone VoIP espera encontrar no servidor TFTP um ficheiro de configuração cujo nome é o seu endereço MAC. Estas configurações únicas podem ser a extensão, nome de apresentação do visor, toque, entre outras.

2.4.2 FTP

O FTP ou *File Transfer Protocol* é um protocolo especificado pela RFC959 [Postel85], que proporciona uma forma rápida de transferência de ficheiros sobre uma rede IP. Para o seu funcionamento é necessária uma componente servidora e uma componente cliente, que comunicam entre si sob um modelo de ligação Servidor-Cliente [NE96]. O cliente só consegue efectuar uma ligação ao servidor se tiver credenciais (conjunto utilizador/palavra-passe) ou então em modo anónimo, se o mesmo o permitir.

A utilização de um servidor FTP para provisionar terminais IP tem apenas algumas diferenças em relação ao método anterior. Uma das diferenças reside na necessidade de utilização de um utilizador para se efectuar a ligação ao servidor FTP. Habitualmente, os telefones já vêm de fábrica preparados com uma conta para

accederem a este tipo de servidores. Sendo assim, apenas é preciso criar uma conta no servidor igual à que vem criada nos terminais e, desta forma, eles acedem sem problemas à sua área de provisionamento, de onde podem descarregar os respectivos ficheiros. Esta área não é mais que a sua pasta pessoal, `/home/terminal_IP`, onde `terminal_IP` corresponde à conta que vem pré-configurada no dispositivo VoIP.

O resto do processo mantém-se idêntico ao anterior, mas em vez dos ficheiros serem transferidos via TFTP, seguem via FTP, usufruindo dos serviços fornecidos pelo TCP.

2.4.3 HTTP

O HTTP (Hypertext Transfer Protocol) é um protocolo de transferência de conteúdo, de texto ou multimédia, muito utilizado na Internet. Este protocolo permite que navegadores *web* descarreguem ficheiros ou uma página de modo que esta informação seja revelada ao utilizador. Este protocolo é especificado pelo *Internet Engineering Task Force* sob a RFC2616. [Fielding99]

Nos terminais, para os quais se centram os objectivos da plataforma a desenvolver, alguns deles, suportam a utilização de TFTP em conjunto com o HTTP. Numa primeira fase, o funcionamento passa por fazer o terminal IP descarregar o *firmware* e o primeiro ficheiro de configuração por TFTP e, a partir daí tudo o resto pode ser feito via HTTP. O ficheiro de configuração inicial é que dita os procedimentos a seguir, ou seja, se as restantes configurações são descarregadas por TFTP ou por HTTP.

Os terminais VoIP do IPL que suportam estes dois métodos de provisionamento são o Polycom 330 e o Linksys SPA942.

Alguns telefones fazem uso de um misto entre os protocolos TFTP e HTTP para conseguirem realizar o seu provisionamento.

Numa solução típica de provisionamento, o *firmware* e o ficheiro de configuração principal são descarregados usando TFTP. No ficheiro de configuração descarregado existem referências para ficheiros que o telefone vai descarregar, como por exemplo

agendas, ficheiros de linguagem, sons, imagens, entre outros. O telefone passa então a descarregar estes restantes ficheiros usando HTTP. Uma grande vantagem do uso deste método mede-se com a existência do controlo de transporte, uma vez que este utiliza o protocolo TCP. Há todo um outro conjunto de vantagens associadas ao HTTP, como por exemplo, a acessibilidade através de um *browser*, *uploads* fáceis através de contas FTP ou mesmo através de contas em Linux acessíveis utilizando SSH. Existem, no entanto, telefones que usam unicamente o HTTP como método de provisionamento. Entre os raros exemplares encontram-se alguns *softphones*. Todos os *softphones* conhecidos que implementam esta função são pagos, por isso, não foi possível efectuar o estudo a este nível.

2.4.4 DHCP

Um serviço DHCP (*Dynamic Host Configuration Protocol*) é responsável pela atribuição dinâmica de configurações de rede a qualquer dispositivo que a ela se ligue. Num servidor de DHCP, para além das configurações de IP, é possível atribuir da mesma forma outras configurações, entre as quais a opção 66, onde se define a localização de um servidor TFTP.[Yang10]

É um dos componentes essenciais numa plataforma de provisionamento de terminais VoIP, pois, sem DHCP, os telefones não teriam capacidade de conhecer a localização das configurações, sendo necessário configurá-las manualmente.

Quando se configura a opção 66 no servidor DHCP, normalmente também se especifica a imagem, ou os ficheiros a carregar pelos dispositivos que se ligam ao servidor TFTP. Esta configuração é a opção 67. No caso dos telefones IP existentes no Instituto Politécnico de Leiria, esta operação não será necessária, uma vez que estes já vêm preparados de fabrica para descarregar determinados ficheiros. [Connor04]

A figura 2.10 representa as várias fases de provisionamento de um terminal IP.

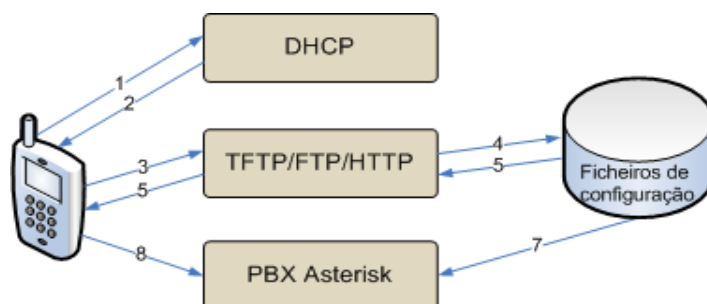


Figura 2.10 - fases do provisionamento

2.4.4.1 Configuração de um serviço de DHCP

Na figura 2.11 é apresentado um exemplo de configuração de um serviço típico de DHCP em sistemas operativos Linux.

Analizando a figura seguinte a primeira instrução indica qual a interface de rede pela qual o serviço DHCP vai responder. A instrução seguinte define a constante **boot-server** como *string*, para mais tarde lhe ser atribuído um valor. Esta informação é passada ao terminal na altura de arranque, sendo que este saberá o que fazer com ela.

O “Bloco de subrede” serve para definir uma rede que o DHCP vai utilizar para atribuir endereços. Dentro deste bloco ainda haverá uma gama de endereços a atribuir com a instrução **range dynamic-bootp 192.168.10.50 192.168.10.150**. As quatro instruções seguintes definem qual a porta de ligação (*gateway*), a máscara de rede, o servidor de DNS e de NTP (sincronização fuso horário) que o serviço de DHCP vai atribuir.

Seguem-se as instruções de concessão do endereço IP. No exemplo dado na figura, é definido um tempo por omissão de 300s (**default-lease-time 300**). Se o cliente solicitar mais tempo de concessão o máximo que poderá ser atribuído é 3600s (**max-lease-time 3600**).

Dentro da secção **group {}** é possível agrupar configurações e parâmetros por cada tipo de dispositivo, por exemplo para todos os terminais de uma determinada marca/modelo. Para isso é preciso agrupá-la, através dos seus endereços MAC.

Como se pode ver na figura, para aquele grupo foi especificado o servidor TFTP (**option tftp-server-name "192.168.9.244"**) e a *string* a passar para os terminais aí incluídos (**option boot-server "ftp://PlcmSpIp:PlcmSpIp@192.168.9.244;"**).

```
# Interface de rede
DHCPDARGS=eth1;

# Activar a opção 66
option boot-server code 66 = string;

# Bloco de subrede
subnet 192.168.10.0 netmask 255.255.255.0 {

range dynamic-bootp 192.168.10.50 192.168.10.150

# Porta de ligação e Máscara de rede
option routers 192.168.10.254;
option subnet-mask 255.255.255.0;

# DNS e Servidor de sincronização de fuso horário
option domain-name-servers 192.168.10.10;
option ntp-servers 192.168.10.1;

# Concessão do IP
default-lease-time 300;
max-lease-time 3600;

group {
option tftp-server-name "192.168.10.1";
option boot-server "ftp://senha:utilizador@192.168.10.1";
host Poly709001 { hardware ethernet 00:04:f2:1e:ac:44; }
}

} Fim do bloco
```

Figura 2.11 - Exemplo de configuração de um servidor DHCP

Todas estas configurações são efectuadas num ficheiro de configuração denominado de **dhcpd.conf**, localizado normalmente em **/etc** num servidor Linux. [Connor04]

2.5 TR-069 CPE WAN Management Protocol

O TR-069 ou CPE WAN Management Protocol (CWMP) é um protocolo de provisionamento via comunicações ADSL. Foi desenvolvido pelo Broadband Forum,

com o objectivo de uniformizar a comunicação entre um equipamento localizado no fim da linha de comunicações (CPE¹), e um equipamento de gestão localizado nas instalações do operador (ACS²).[BF07]

O Broadband Forum é um consórcio de aproximadamente 200 empresas da indústria das comunicações, computação, redes e operadores de telecomunicações. Foi estabelecido em 1994, inicialmente, com o nome de ADSL Forum. Mais tarde, viria a adoptar a designação de Broadband Forum. [BDF10]

No domínio das telecomunicações, um *Customer Premises Equipment* (CPE) é uma designação atribuída aos dispositivos terminais que se encontram nas instalações do cliente. Estes dispositivos são parte integrante do sistema fornecido por um operador, efectuando normalmente a terminação da linha de comunicações. Alguns exemplos destes equipamentos podem ser, receptores da TV cabo, telefones, modems, *routers* (ADSL ou cabo), ou mesmo terminais de voz sobre IP.[BF07]

Um ACS ou Auto Configuration Server é um serviço que é responsável por gerir os dispositivos CPE que suportem o protocolo TR-069. Este serviço não é mais que uma aplicação a correr num servidor do lado do operador de telecomunicações, que poderá ter como principais funções: configuração, monitorização e gestão de todo o parque de terminais TR-069. A principal função de um ACS é supostamente ter a capacidade de configurar um determinado equipamento sem a intervenção do utilizador final. As únicas intervenções do utilizador serão tarefas tão simples como, retirar o equipamento da caixa, ligar à linha do operador e ligar o botão de energia. [BF07]

A figura 2.12 demonstra a ligação que existe entre o servidor ACS e os dispositivos CPE através do protocolo CWMP. Pode-se verificar, nesta figura, que um operador do centro de atendimento (*call center*) poderá resolver problemas num dispositivo localizado em casa do cliente (CPE).

¹CPE ou *Customer Premises Equipment* – é o equipamento localizado no fim de linha de comunicação, normalmente dentro do recinto do cliente.

²ACS ou *Auto Configuration Servers* – é a componente servidor de provionmento do protocolo TR-069.

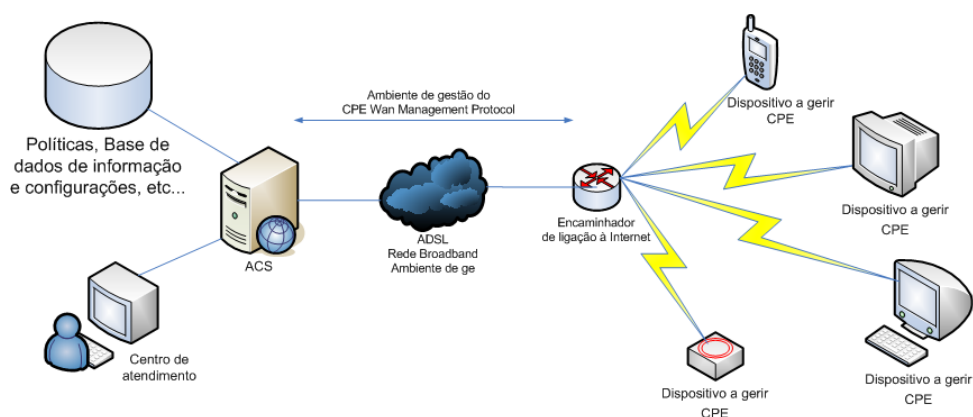


Figura 2.12 - Ambiente de gestão utilizando o protocolo CWMP

O CWMP, para além de especificar o protocolo que interliga o ACS ao CPE, também prevê diversas funcionalidades, tais como:

- Configuração automática;
- Gestão do *firmware* e *software* do equipamento;
- Estado e monitorização de desempenho;
- Diagnósticos.

Em termos de arquitectura, o *CPE Wan Management Protocol*, utiliza protocolos universais, tais como o HTTP, SOAP, e o RPC, juntando os seus próprios componentes para assim criar a sua arquitectura.

O SOAP é um protocolo baseado em XML que permite a troca de mensagens através de HTTP, tendo nascido como uma recomendação do consórcio *World Wide Web* (W3C¹). Nesta arquitectura o SOAP [Baker04] trata de codificar as mensagens RPC de forma a transmiti-las via HTTP.

¹ W3C – O consórcio *The World Wide Web* (W3C) desenvolve tecnologias para a *web* com a finalidade de conseguir-se usar o o maior potencial possível da mesma. O sítio *web* oficial é www.w3.org.

A Chamada de Procedimento Remoto (RPC - Remote Procedure Call) é um protocolo que permite a uma aplicação requisitar um serviço ou um processo a outra aplicação normalmente localizada noutro computador. No fundo trata-se de uma comunicação entre processos. Como o RPC não trabalha sobre o protocolo HTTP, a norma TR-069, utilizou o SOAP para o encapsular conseguido assim a comunicação entre os processos RPC. [Srinivasan95]

A camada de HTTP será utilizada para transportar as mensagens SOAP entre o CPE e o ACS, agindo neste caso o CPE como cliente HTTP e o ACS como Servidor HTTP. Pode-se dar também o caso da comunicação ser iniciada pelo ACS, agindo neste caso o ACS como cliente HTTP e o CPE como servidor.[BF07]

A recomendação CWMP também especifica componentes de segurança, daí a necessidade da camada SSL/TLS[Dierks99]. Aqui é providenciada confidencialidade da informação transmitida, autenticação e integridade, e ainda a possibilidade de utilização de autenticação baseada em certificados entre o CPE e o ACS.[BF07]

Existe uma alternativa de autenticação do CPE na camada HTTP que consiste em utilizar um algoritmo *shared-secret*. A autenticação baseada em *shared-secret* passa por serem cifrados blocos de informação de pequena quantidade. Estes blocos são cifrados com uma chave comum aos dois interlocutores, e a mesma serve para cifrar e decifrar. A desvantagem deste método é que ambos os interlocutores têm de possuir ou acordar a chave previamente à transmissão de dados, caso contrário, tem de a transmitir através de uma canal inseguro.

A recomendação TR-069 é complementada com outras especificações que tratam por exemplo do provisionamento de computadores pessoais, terminais VoIP, servidor domésticos de multimédia, entre outros. Na figura 2.13 é apresentado um esquema que mostra a interligação dos alguns protocolos incorporados no TR-069.

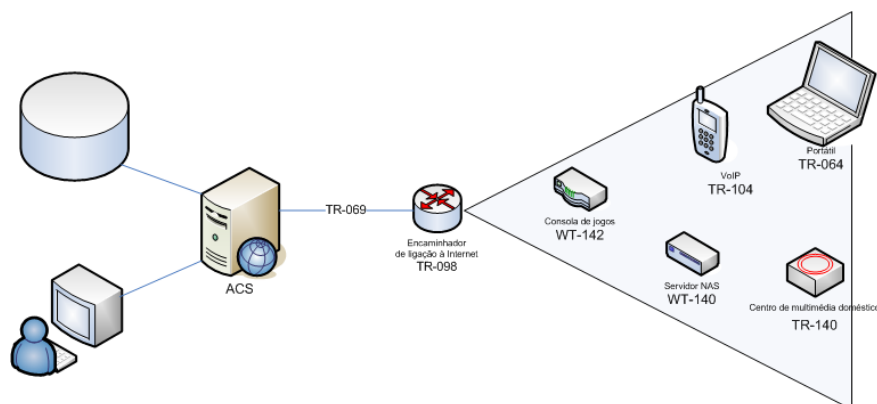


Figura 2.13 - Alguns protocolos incorporados no TR-069

2.5.1 Síntese

Neste capítulo foram identificados termos, protocolos e tecnologias que estão ligados à área de provisionamento de dispositivos de voz sobre IP.

Apresentou-se de forma sumária o tema Voz Sobre IP, os protocolos de sinalização mais usados por esta tecnologia, os dispositivos que compõem este sistema e ainda a ferramenta Asterisk, uma implementação livre de IPPBX.

Definiu-se o termo “provisionamento” bem como formas e métodos de implementação, identificando vários componentes de rede essenciais, como o caso do DHCP, TFTP, FTP ou HTTP.

Embora ainda limitadas a preencher os requisitos necessários, apresentaram-se as ferramentas livres de provisionamento, a *End Point Manager* e a VGPS.

Não se fez mais que uma breve descrição do VGPS, uma vez, que o seu desenvolvimento está estagnado já à bastante tempo.

O *End Point Manager* apesar incorporar todas as marcas que o IPL possui, não se revelou uma boa opção devido a ser uma ferramenta que está incluída numa distribuição Asterisk, não funcionando em separado.

Relativamente a soluções comerciais, elaborou-se um quadro que compara essencialmente os telefones IP suportados por cada uma.

Por fim, neste capítulo apresentou-se o estudo feito ao protocolo de provisionamento *CPE WAN Management Protocol*.

3 Terminais VoIP

Neste capítulo será descrito o estudo efectuado individualmente a cada um dos terminais VoIP existentes no IPL. Este estudo caracterizará o processo de provisionamento necessário para colocar um destes dispositivos em funcionamento numa rede VoIP. Será também dada ênfase a personalizações do toque de chamada, imagem de logotipo, agenda, idioma e teclas de atalho.

A diferença em cada um dos modelos de telefones IP existentes no IPL, em termos de configuração, personalização e utilização, leva a que o estudo, efectuado individualmente a cada terminal, seja essencial. Só conhecendo as características aprofundadas de cada um se pôde avançar para a idealização de uma arquitectura e posteriormente à construção de um protótipo.

Embora tenham sido estudadas as mesmas funcionalidades para cada terminal estas na prática implementam-se de forma distinta, variando de telefone para telefone.

3.1 Polycom SoundPoint IP330

O modelo SoundPoint IP 330 é bastante completo em termos de opções e funcionalidades. A sua configuração pode ser provisionada de várias formas, nomeadamente FTP, TFTP e HTTP.

A figura 3.1 apresenta o telefone Polycom em questão.



Figura 3.1 - Terminal Polycom SoundPoint IP330

Por omissão, este telefone vem configurado de fábrica para obter o endereço IP automaticamente (através de servidor DHCP) e as configurações de provisionamento através de FTP. Assim, na plataforma, o método de provisionamento preferencial para este modelo de telefone será via FTP, pois desta forma será apenas necessário ligar o telefone na rede para o provisionar, isto é, sem necessidade de configurações prévias em cada telefone.

Para aceder ao servidor FTP e descarregar as configurações, o telefone Polycom 330 é munido de um utilizador e palavra-passe pré-configurado, logo, no servidor de provisionamento será necessário criar um utilizador e uma palavra-passe igual.

Cada vez que o telefone é ligado efectua um pedido de DHCP em *broadcast*¹ à rede. O servidor DHCP responde com um endereço de rede IP e um *boot-server*², indicando o caminho para o servidor de FTP, incluindo o utilizador e palavra-passe.

A configuração *boot-server*, no servidor DHCP, será do tipo: **option boot-server "ftp://utilizador:senha@192.168.0.100"**, sendo que o **utilizador** e **senha** convém serem iguais aos que vêm configurados de fabrica, no terminal IP, de forma a se economizar trabalho. Ainda na fase de arranque, o próximo passo do terminal será ligar-se ao servidor FTP e aceder ao seu directório pessoal (**/home/polycom**).

¹*Broadcast*, – Envio de pacotes para todos os dispositivos de uma rede.

²*Boot-server* – Configuração num servidor DHCP que indica ao cliente o caminho de uma imagem de arranque.

Para o terminal Polycom conseguir identificar onde ir buscar cada ficheiro, quer seja de configuração, toque, agenda ou outro qualquer deve existir, no seu directório pessoal, a estrutura apresentada na figura 3.2.

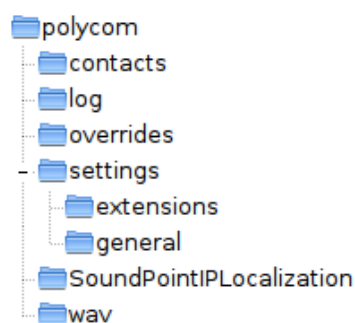


Figura 3.2 - Árvore de directórios para o Polycom 330

A tabela 3.1 descreve a função de cada pasta que está localizada no directório pessoal do terminais IP Polycom – **/home/Polycom**.

Directório/pasta	Função
<i>contacts</i>	É usada para armazenar todos os ficheiros que digam respeito à lista de contactos telefónicos.
<i>log</i>	É a pasta onde são armazenados todos os relatórios de monitorização, enviados pelo telefone, cada vez que se faz uma alteração na sua configuração ou quando este é reiniciado.
<i>overrides</i>	A pasta <i>overrides</i> tem a finalidade de guardar ficheiros de configuração, sem que o envio seja feito sempre que se proceda a qualquer alteração de configuração ou personalização no telefone.
<i>settings</i> <i>extensions</i> <i>general</i>	Estas pastas contêm todos os ficheiros pertencentes à configuração do telefone propriamente dita.
<i>SoundPointIPLocalization</i>	Contém os ficheiros ou pastas para configuração do idioma do telefone.
<i>wav</i>	A pasta <i>wav</i> serve para guardar os toques personalizados dos telefones Polycom.

Tabela 3.1 - Função dos directórios contidos em **/home/Polycom**

Mesmo depois de configurado a primeira vez, o Polycom arrancará sempre a partir do endereço de *boot-server*, na raiz do seu directório pessoal. Neste processo de arranque é verificado se existem versões do *firmware* mais recentes e, se for o caso,

efectuada a sua actualização. Portanto, para efectuar uma actualização basta copiar para esse directório o *firmware* mais actualizado. O *firmware* é constituído por um conjunto de ficheiros **xx_sip.ld** e um ficheiro **sip.ver**, que se encontram num pacote compactado **spip_ssip_vvx_3_1_3RevB_release_sig_combined.zip** disponível na página *web* da Polycom¹.

O diagrama ilustrado na figura 3.3 caracteriza o arranque do terminal IP Polycom 330, onde se pode verificar que este se pode resumir a cinco fases.



Figura 3.3 - Fase de arranque do Polycom 330

As configurações dos telefones IP da Polycom são feitas com base em ficheiros programados na linguagem *eXtensible Markup Language* (XML).

A linguagem XML, num determinado nível, é um protocolo para conter e gerir informação. Noutro nível, é uma família de tecnologias que tem a capacidade de fazer tudo em relação à formatação de documentos, de forma a filtrar dados. Num nível mais elevado, é uma filosofia de manipulação de informação focando a máxima utilidade e flexibilidade, aperfeiçoando os dados na sua forma mais pura e estruturada.[Ray03]

Na figura 3.4 pode-se visualizar o conteúdo do ficheiro **000000000000.cfg**, que não é mais que o ficheiro principal de configuração, usado como modelo de configuração. Este ficheiro deve ser renomeado para o endereço MAC do terminal IP, ficando com a seguinte estrutura: **0004f200106c.cfg**. Contudo é recomendável haver um ficheiro com

¹http://www.polycom.com/support/voice/soundpoint_ip/soundpoint_ip330_320.html

a designação: **000000000000.cfg**, para carregar configurações por omissão, no caso de algum telefone não encontrar o endereço MAC correspondente.

```

                                000000000000.cfg
<?xml version="1.0" standalone="yes"?>
<!-- Default Master SIP Configuration File -->
<!-- edit and rename this file to <Ethernet-address>.cfg for
each phone. -->
<!-- $Revision: 1.14 $ $Date: 2005/07/27 18:43:30 $ -->
< APPLICATION APP_FILE_PATH="sip.ld" CONFIG_FILES="phone1.cfg, sip.cfg" MISC_FILES="" LOG
FILE DIRECTORY="" OVERRIDES_DIRECTORY="" CONTACTS_DIRECTORY=""
LICENSE_DIRECTORY="" />

```

Figura 3.4 - Conteúdo do ficheiro 000000000000.cfg

O ficheiro de configuração **000000000000.cfg** é ainda responsável pela ligação entre o telefone e os restantes ficheiros de configuração, pela personalização e pela atribuição de extensão telefónica. É neste código XML que é definido o número de ficheiros, e qual a ordem pela qual o telefone os irá procurar. Neste pedaço de código o ficheiro **phone1.cfg** é lido primeiro que o **sip.cfg**, sendo que, as configurações que prevalecem são as do primeiro ficheiro. Assim, devem-se colocar os ficheiros com configurações mais específicas à esquerda e os mais genéricos (configurações por omissão, por exemplo) à direita. Na figura 3.5 é apresentado um ficheiro de configuração **0004f200106c.cfg** onde se pode comprovar que primeiro são carregados os ficheiros de configuração da extensão (**x500.cfg**; **app_config_330.cfg**.) e depois as configurações genéricas (**phone1.cfg**; **sip.cfg**).[PL10]

```

                                0004f200106c.cfg
<?xml version="1.0" standalone="yes"?>
<!-- Default Master SIP Configuration File-->
<!-- Edit and rename this file to <Ethernet-address>.cfg for each phone.-->
<!-- $Revision: 1.14 $ $Date: 2005/07/27 18:43:30 $ -->
< APPLICATION APP_FILE_PATH="sip.ld" CONFIG_FILES="/settings/extensions/x500.cfg,
/settings/app_config_330.cfg, /settings/general/phone1.cfg, /settings/general/sip.cfg" MISC_FILES=""
LOG_FILE_DIRECTORY="/log" OVERRIDES_DIRECTORY="/overrides"
CONTACTS_DIRECTORY="/contacts"/>

```

Figura 3.5 - Conteúdo do ficheiro 0004f200106c.cfg

Quanto aos ficheiros **x500.cfg** e **app_config_330.cfg** foi referido que tratam das configurações de ligação do telefone ao IPPBX, ou seja, indicam a extensão e credenciais a usar e o servidor onde se registam. Estes dois exemplos, foram criados para atribuir a extensão 500, já previamente criada no Trixbox, ao terminal IP Polycom 330 com endereço MAC 0004f200106C.

A figura 3.6 representa a atribuição de vários parâmetros ao terminal IP. O campo *displayName* corresponde ao nome que irá ser apresentado no visor do telefone, o campo *address* corresponde à extensão telefónica e o *userId* equivale ao utilizador criado no Asterisk que, tipicamente, é igual à extensão. O campo *password* corresponde à palavra-passe associada à extensão, o campo *ringType* estabelece o tipo de toque pré-definido para o utilizador do telefone, e por último, o campo *lineKeys* especifica o número de linhas que o telefone tem à disposição (*line1*, *line2*,...), que neste modelo são duas.

```

                                x500.cfg
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Per-phone configuration in this file -->
<reginfo>
<reg
reg.1.displayName="Zé Mário" reg.1.address="500" reg.1.auth.userId="500"
reg.1.auth.password="senhaSecreta" reg.1.ringType="15" reg.1.lineKeys="2"
/>
</reginfo>

```

Figura 3.6 - Ficheiro de configuração da extensão

O **app_config_330.cfg** contém configurações adicionais ao ficheiro que vem fornecido de fábrica, o **sip.cfg**, e é o primeiro a ser carregado quando o telefone arranca. De seguida são carregadas as instruções do **sip.cfg**, não sendo estas sobrepostas em caso de repetição de comandos. Na figura 3.7 é possível visualizar as configurações que indicam qual o servidor onde o telefone tem de se registar.

```

app_config_330.cfg
<server voIpProt.server.1.address="192.168.0.100"/>
<SIP>
<outboundProxy voIpProt.SIP.outboundProxy.address="192.168.0.100"/>
</SIP>

```

Figura 3.7 - Conteúdo do ficheiro app_config_330.cfg

De forma a resumir algumas das características de provisionamento do telefone IP Polycom 330, é apresentada a tabela 3.2.

Método de provisionamento	FTP (predefinido), TFTP e HTTP
Utilizador/Palavra-passe para FTP	PlcmSpIp/PlcmSpIp
Teclas para reiniciar o telefone	Volume-, Volume+, Em espera e Mão livres (e aguardar uns segundos)
Ficheiros com configurações de fábrica	phone1.cfg sip.cfg
Ficheiro de configuração genérico	app_config_330.cfg
Ficheiros de configuração específicos (extensão, senha, nome de apresentação)	xExtensão.cfg endereçoMAC.cfg

Tabela 3.2 - Resumo de funções do Polycom 330

As secções que se seguem, descrevem as personalizações possíveis de efectuar remotamente a este terminal IP.

3.1.1 Personalização do idioma

Como o telefone Polycom SoundPoint IP 330 vem configurado de fábrica com o idioma na língua inglesa, nesta secção, será apresentada uma forma de carregar o dicionário de língua Portuguesa e colocá-lo activo, por omissão.

No primeiro passo, terá que se copiar o conteúdo da pasta **SoundPointIPLocalization** do ficheiro **spip_ssip_vvx_3_1_3RevB_release_sig_combined.zip**¹ para o directório do servidor

¹Este ficheiro acompanha o telefone aquando da sua compra, ou então pode-se descarregar de <http://support.polycom.com/PolycomService/support/us/support/voice/index.html>

de provisionamento `/home/polycom/SoundPointIPLocalization`. Esta pasta contém diversos ficheiros relativos aos idiomas que a Polycom suporta.

O passo seguinte é alterar o ficheiro de configuração `app_config_330.cfg` de modo a carregar o idioma Português, por omissão, e também, deixar acessível outros idiomas para que o utilizador os possa escolher na configuração do terminal.

O código a acrescentar no ficheiro `app_config_330.cfg` é o apresentado na figura 3.8.

app_config_330.cfg

```
(...)
<localization>
<multilingual>
<language lcl.ml.lang="Portuguese_Portugal">
  <menu
    lcl.ml.lang.menu.1="Chinese_China" lcl.ml.lang.menu.2="English_United_Kingdom"
    lcl.ml.lang.menu.3="Norwegian_Norway"
  />
</language>
</multilingual>
</localization>
```

Figura 3.8 - Conteúdo acrescentado ao ficheiro `app_config_330.cfg`

3.1.2 Personalização do toque

A personalização do toque permitirá ao administrador alterar o som pré-definido de qualquer telefone, bem como, a colecção de toques disponíveis ao utilizador.

Como requisitos obrigatórios, os ficheiros de áudio têm de estar no formato WAV (Waveform Audio File Format), serem mono e terem uma frequência de 8kHz, caso contrário, o telefone poderá não os conseguir reproduzir.

O código XML, a acrescentar no ficheiro criado para configuração/personalização do Polycom 330, ou seja, no `app_config_330.cfg`, está apresentado na figura 3.9. Analisando o código, verifica-se que cada atributo `saf` corresponde a um toque, onde é preciso indicar o nome e o caminho. O Polycom 330 suporta apenas onze toques, logo, a partir de `saf.12` deve-se deixar em branco.

```

app_config_330.cfg
(...)
<sampled_audio saf.1="/wav/campainha.wav" saf.2="/wav/mozart_theme.wav"
saf.3="/wav/carlosparedes_theme.wav" saf.4="/wav/polycom_dt_300.wav_330.wav" saf.5="" saf.6="" saf.7=""
saf.8="" saf.9="" saf.10="" saf.11="" saf.12=""/>

```

Figura 3.9 - Conteúdo acrescentado ao ficheiro app_config_330.cfg

3.1.3 Personalização da imagem de logotipo

Para carregar um logotipo ou simplesmente uma imagem no terminal Polycom 330, é necessário garantir algumas características a nível gráfico. A imagem tem de ser monocromática, com formato BMP (BITMAP¹) e com dimensões que não ultrapassem os 102 por 23 pixels.

Supondo que a imagem que se pretende carregar tem o nome de **P330_logotipo.bmp**, então os atributos que se devem configurar são os apresentados na figura 3.10. Note-se que apenas é necessário dar o nome do ficheiro excluindo a extensão (**.bmp**).

```

app_config_330.cfg
(...)
<bitmaps>
<IP_330
  bitmap.IP_330.61.name="P330_logotipo"
</bitmaps>
<Animations>
<IP_330>
  <IDLE_DISPLAY ind.anim.IP_330.29.frame.1.bitmap="P330_logotipo"
  ind.anim.IP_330.29.frame.1.duration="0"/>
</IP_330>
</Animations>

```

Figura 3.10 - Conteúdo acrescentado ao ficheiro app_config_330.cfg

A configuração da imagem a carregar é feita no parâmetro **bitmap.IP_330** na posição 61 pois, ao analisar o ficheiro **sip.cfg**, verificou-se que esta era a primeira posição livre.

¹Windows bitmap é um formato de imagem sem compressão e de simples complexidade, desenvolvido pela Microsoft.

3.1.4 Agenda de contactos

Uma das funcionalidades igualmente necessárias à plataforma de provisionamento é a possibilidade de inserir uma agenda de contactos, com a finalidade dos utilizadores consultarem uma agenda comum a toda a instituição.

Mais uma vez se irá recorrer a ficheiros XML para guardar a informação a enviar para os telefones. O ficheiro de agenda é construído com as *tags*¹ `<fn>Nome</fn>`, `<ct>Extensão Telefónica</ct>` e `<sd>Ordem de listagem dos contactos</sd>`. Na figura 3.11 pode-se verificar com mais detalhe o ficheiro XML com alguns contactos de exemplo.

```

                                000000000000-directory.xml
<?xml version="1.0" standalone="yes"?>
<directory>
  <item_list>
    <item><fn>Angelina</fn><ct>500</ct><sd>1</sd></item>
    <item><fn>Ze Nando</fn><ct>501</ct><sd>2</sd></item>
    <item><fn>Soraia Sa</fn><ct>502</ct><sd>3</sd></item>
  </item_list>
</directory>

```

Figura 3.11 - Ficheiro da agenda de contactos

É possível utilizar o ficheiro `000000000000-directory.xml` que vem fornecido no pacote, já mencionado – `spip_ssip_vvx_3_1_3RevB_release_sig_combined.zip`. Depois de construído o ficheiro de agenda, este deve ser copiado para a pasta de contactos do Polycom, no servidor de provisionamento – `/home/polycom/contacts`.

O ficheiro `000000000000-directory.xml` pode ser renomeado para o endereço MAC do telefone IP, `MAC-directory.xml`, afectando desta forma apenas este terminal. Para a agenda ser carregada em todos os terminais IP Polycom basta deixar o nome, por omissão.

¹*Tag*, é uma marcação ou etiqueta que define um significado para tipos de dados.

3.2 Snom 360

O telefone IP Snom 360 é um dos modelos que o Instituto Politécnico de Leiria possui no seu parque de dispositivos VoIP. Na figura 3.12 encontra-se uma imagem deste modelo de terminal IP.



Figura 3.12 - Telefone Snom 360

O Snom 360 é um telefone bastante completo, à semelhança do Polycom 330. Apresenta um vasto leque de opções, sendo algumas delas até superiores ao Polycom 330. Devido ao seu ecrã de grandes dimensões e à quantidade de teclas de atalho (ou de marcação rápida), revela-se bastante versátil, pois pode servir tanto a um utilizador comum como a uma operadora. Este terminal IP também permite ser expandido com painéis, constituídos por múltiplos botões/teclas de atalho, que servem para marcação rápida de extensões.

À semelhança dos Polycom e outros equipamentos VoIP, o Snom 360 tem incorporado um servidor *web* que dá acesso a uma página de configuração simples e intuitiva.

A vantagem de ter um ecrã grande é a possibilidade de visualizar várias informações de uma só vez, como por exemplo, visualizar múltiplos contactos da agenda enquanto se pesquisa.

Os terminais IP Snom 360 existentes no IPL, possuem a versão do seu *firmware* ainda na versão 6. Nesta versão ainda não são suportadas configurações recorrendo a ficheiros de marcação XML, uma vez que só admite ficheiros com um formato proprietário.

Assim, para se usufruir de todas as vantagens e simplicidade da linguagem XML, devem-se actualizar os terminais para a versão 7. Nesta versão, o sistema operativo deixa de ser proprietário para dar lugar a um baseado em Linux, permitindo o provisionamento, através de TFTP, HTTP e HTTPS.

3.2.1 Actualização do *firmware* para versão 7

A actualização para a versão 7 é essencial, devido às vantagens da linguagem XML, mas também para haver uma continuidade de actualizações que estagnariam com a versão 6. A transição da versão 6 para a 7 não é muito directa, sendo necessário, carregar um *firmware* intermédio – **snom360-from6to7-7.3.14-bf.bin**, que pode ser descarregado da página do fabricante¹. Após a instalação deste *firmware*, o terminal IP ficará com uma funcionalidade bastante necessária que é a de permitir fazer actualizações automáticas. A instalação deste *firmware* intermédio é feita a partir da sua página *web* de configuração, onde existe um campo que permite o envio do ficheiro de actualização.

Depois do envio do ficheiro de actualização, o telefone é reiniciado e fica apto a receber o *firmware* da versão 7, ou superiores.

Para se proceder à actualização, deve-se ter um servidor TFTP onde colocar o ficheiro de *firmware* mais actual e os ficheiros de configuração. Ao arrancar, o Snom 360 vai primeiro procurar no servidor de TFTP o ficheiro de configuração, que por sua vez lhe identificará qual o *firmware* que ele deveria possuir. No caso de ser inferior, o telefone é automaticamente actualizado.

¹Firmware de transição da versão 6 para a 7 - <http://fox.snom.com/update6to7/snom360-from6to7-7.3.14-bf.bin>, sítio *web* disponível em Abril de 2010.

O ficheiro XML de configuração deverá ter o nome **snom360.xml** incluindo o código que indica qual o *firmware* que o telefone tem de ter, de acordo com a figura 3.1.3.

```
snom360.xml
(...)
<firmware perm=""> http://192.168.0.100/snom/snom360-from6to7-7.3.14-bf.bin </firmware>
```

Figura 3.13 - *Firmware* do telefone

O terminal IP, quando é ligado, reconhece qual é o servidor de TFTP porque, todas as configurações de rede são atribuídas por um servidor DHCP que, obrigatoriamente, tem de existir na rede.

3.2.2 Correio de voz

Como já foi referido, o Asterisk permite aos seus utilizadores acederem ao seu correio de voz, marcando ***97**. O Snom 360 possui uma tecla própria para ligar para o correio de voz, no entanto, esta tecla pode ser configurada para qualquer outra função.

A figura 3.14 apresenta o código necessário para configurar a tecla de correio de voz para todas as identidades.

```
snom360-MAC.xml
(...)
<user_mailbox idx="1" perm="R">*97</user_mailbox>
<user_mailbox idx="2" perm="R">*97</user_mailbox>
<user_mailbox idx="3" perm="R">*97</user_mailbox>
...
<user_mailbox idx="12" perm="R">*97</user_mailbox>
```

Figura 3.14 - Programação da tecla de correio de voz

Para além das características, já abordadas, que este telefone tem direccionado a operadoras, a possibilidade de 12 identidades é mais uma, isto é, este terminal IP admite registar-se com 12 contas SIP, quer seja ou não no mesmo servidor.

3.2.3 Personalização das teclas de atalhos

As 12 teclas, disponibilizadas pelo terminal IP Snom 360, podem ser configuradas para diversas funções. As mais usuais são as de marcação rápida, ou seja, o operador do telefone pode fazer chamadas telefónicas apenas pressionando um único botão de atalho.

Outra função que se pode atribuir a estas teclas, é a troca de identidade, que pode ser muito útil a uma operadora que faça a gestão de várias linhas telefónicas. Na página *web* deste telefone, que se pode aceder inserindo o seu endereço IP num navegador de Internet, é possível efectuar, facilmente, configurações das teclas de atalho. De qualquer forma, o objectivo desta dissertação é apresentar as variadas possibilidades de personalizar configurações no terminais IP, remotamente. Assim, as linhas de código para configurar as teclas de atalho são apresentadas na figura 3.15.

```
snom360-MAC.xml
(...)
<fkey idx="4" context="active" perm=""></fkey>
<fkey idx="5" context="active" perm="">line</fkey>
<fkey idx="6" context="active" perm="">speed 501</fkey>
<fkey idx="7" context="active" perm="">speed 502</fkey>
...
<fkey idx="11" context="active" perm="">speed 506</fkey>
```

Figura 3.15 - Configuração das teclas de atalho

Analisando o código anterior, confirma-se a possibilidade de configurar várias funções em cada tecla. A função **speed 501** permite a ligação rápida para essa extensão. A função **line** permite associar uma tecla a uma entidade ou linha. Desta forma, consegue-se rapidamente puxar a linha de uma determinada identidade ou mesmo saber qual a linha pela qual está a chegar uma chamada, uma vez que existe um LED (*Light Emitter Diode*) junto a cada tecla de atalho, e esta pisca até ser atendida.

Existem muitas outras acções possíveis a atribuir às teclas de atalho. Na tabela 3.3 são apresentadas as mais utilizadas.

Acção	Função
LINE	Permite associar uma tecla a uma entidade ou linha.
SPEED DIAL	Possibilita ligação rápida para determinada extensão.
CALL LIST	Permite visualizar um histórico das chamadas.
MUTE	Silencia uma chamada em curso.
CONFERENCE	Iniciar uma conferência com uma chamada a decorrer.
RECORDER	Permite gravar uma chamada telefónica.
DIRECTORY SEARCH	Possibilita pesquisar numa <i>Active Directory</i> enquanto decorre uma chamada.

Tabela 3.3 - Acções para botões de atalho¹

Quanto à configuração das teclas do módulo de expansão, esta é feita da mesma forma, pois as teclas são a continuação das do telefone, logo, no código XML é preciso apenas continuar a partir do ultimo atributo **idx="x"**.

Na figura 3.16 encontra-se uma figura de um telefone Snom 360 com módulo de expansão, onde se pode verificar que constitui um terminal que se pode adaptar perfeitamente a uma operadora.



Figura 3.16 - Telefone Snom 360 com módulo de expansão

3.2.4 Personalização da imagem/logotipo

O Snom 360 suporta uma imagem com tamanho máximo de 128 por 64 pixels, monocromática com formato BITMAP e codificada em base64 [Josefsson06].

¹http://wiki.snom.com/Category:Setting:Function_Keys

A imagem deverá ser inserida no ficheiro **idle.xml**, e o código base64 dentro das *tags* **<image>** , no mesmo ficheiro. Para o telefone IP ler o **idle.xml**, este tem de ser invocado no ficheiro de configuração principal, o **snom360-MAC.xml**, conforme se pode visualizar na figura 3.17.

```
snom360-MAC.xml
(...)
<user_xml_screen_url idx="1" perm="">http://192.168.0.100/snom/idle.xml</user_xml_screen_url>
```

Figura 3.17 - Personalização da imagem de logotipo

Normalmente, nos terminais Snom, só o ficheiro principal de configuração é carregado por TFTP, os restantes são via HTTP. Portanto, neste caso, o **idle.xml** será procurado por HTTP.

3.2.5 Carregar a agenda de contactos

A agenda a disponibilizar ao Snom 360, deve ser construída seguindo a sintaxe XML. Esta poderá ser inserida no **snom360-MAC.xml**, entre as *tags* **<tbook>** conforme a figura 3.18.

```
snom360-MAC.xml
(...)
<tbook e="2">
  <item context="active" type="none" index="1"><name>Ze Nando</name>
  <number>502</number>
  <search></search>
</item>
  <item context="active" type="none" index="2"> <name>Soraia Sa</name>
  <number>503</number>
  <search></search>
</item>
</tbook>
```

Figura 3.18 - Personalizar a agenda de contactos

Analisando o código do exemplo anterior, verifica-se que cada contacto está dentro de uma *tag* **<item>** que consequentemente contém uma *tag* **<name>** e outra **<number>** que correspondem, ao nome e extensão do contacto, respectivamente.

3.2.6 Personalização do toque

Em termos de toques, neste telefone é preciso ter um especial cuidado, já que o Snom 360 tem possibilidade de várias linhas. Assim, se a finalidade deste telefone for para ser usado por uma operadora de atendimento de chamadas, aconselha-se atribuir um toque diferente a cada linha/identidade que esteja registada. Só assim a operadora conseguirá facilmente identificar qual a linha que está a chamar.

Este modelo da Snom suporta toques em formato WAV, com uma amostragem de 8 kHz, mono, e com uma duração de cerca de 7 segundos.

À semelhança de outras configurações, os toques devem ser colocados num directório à parte, acessível por HTTP, isto para o caso do provisionamento ser feito de forma centralizada. Mais uma vez, é no ficheiro **snom360-MAC.xml** que se referencia a localização do ficheiro de toque, como se pode verificar na figura 3.19.

```
snom360-MAC.xml
(...)
<custom_melody_url perm="">http://192.168.0.100/snom/toque_x1.wav
</custom_melody_url> <user_custom idx="1" perm="">http://192.168.0.100/snom/toque_x1.wav
</user_custom> <user_ringer idx="1" perm="">Custom</user_ringer>
```

Figura 3.19 - Personalização do toque

Analisando o código apresentado acima, constata-se que o toque a carregar tem o nome **toque_x1.wav**, e está a ser carregado na identidade 1 (**idx="1"**).

Como se pode confirmar, o ficheiro de toque surge referenciado duas vezes. A primeira serve para carregar o ficheiro no telefone, enquanto que a segunda associa o toque ao utilizador que tiver sessão iniciada no identificador 1, isto é, na primeira conta SIP configurada.

O **custom** serve para carregar o toque por omissão a um utilizador que não tenha toque atribuído, o que acontece se por acaso iniciar sessão um utilizador na linha 2. Este toque **custom** é único no telefone.

3.2.7 Personalização do idioma

Uma grande lacuna do Snom 360 é o facto de não suportar a opção para idioma Português nos seus menus e página *web* de configuração. No entanto, há a possibilidade de se traduzir a partir do conteúdo de um ficheiro de outro idioma. No anexo D encontra-se a tradução que se fez para ser usado na plataforma.

Depois de efectuada a tradução de idioma, devem-se então especificar os ficheiros no já conhecido **snom360-MAC.xml**, como se pode ver na figura 3.20. Recorde-se que estes ficheiros terão que ficar numa área com acesso via HTTP.

```
snom360-MAC.xml
(...)
<web-languages>
  <language url="http://192.168.0.100/snom/web_lang_EN.xml" name="English" $
  <language url="http://192.168.0.100/snom/web_lang_PT.xml" name="Portugues$
</web-languages>
<gui-languages>
  <language url="http://192.168.0.100/snom/gui_lang_EN.xml" name="English"/>
  <language url="http://192.168.0.100/snom/gui_lang_PT.xml" name="Portugues$
</gui-languages>
```

Figura 3.20 - Personalização do idioma

No código acima, não só estão a ser chamados os idiomas da página *web* (**web_lang_PT.xml**), como também o idioma dos menus do telefone (**gui_lang_PT.xml**). Verifica-se também que vão estar disponíveis no telefone IP dois idiomas, Português e Inglês. É ainda de salientar, que o atributo **name** indica o nome que vai aparecer no menu do terminal para escolher o idioma.

Falta então definir qual o idioma que fica, por omissão, disponível ao utilizador, sendo este procedimento feito da forma apresentada na figura 3.21, onde, novamente, uma linha corresponde ao idioma dos menus e a outra à interface *web*.


```
snom360-MAC.xml
(...)
<language perm="RW">Portugues</language>
<web_language perm="RW">Portugues</web_language>
```

Figura 3.21 - Personalização do idioma padrão

3.3 Linksys SPA942

O Linksys SPA942, à semelhança dos dois telefones IP já abordados, é um telefone com características muito direccionados para ambientes empresariais, devido às suas variadas funcionalidades incorporadas. Tem capacidade para suportar duas linhas, ou seja, duas contas SIP, podendo ir até quatro linhas, caso a versão do *firmware* seja superior a 5.1.8 inclusive, que é o caso dos equipamentos existentes no IPL. É composto por quatro teclas de atalho e um ecrã grande, o suficiente para visualizar várias informações. Este telefone é apresentado na figura 3.22.



Figura 3.22 - Telefone IP Linksys SPA942

Em termos de provisionamento, pode ser feito via TFTP, HTTP e HTTPS, sendo que o primeiro é o pré-definido de fábrica.

O provisionamento inicial do Linksys SPA942, ou seja, quando o terminal ainda está com as configurações de fábrica, o ficheiro de configuração que ele procura é o **spa942.cfg**, num servidor TFTP. A indicação do endereço de servidor TFTP e toda a configuração de rede deverão ser atribuídas através de um servidor DHCP.

Depois do Linksys 942 ter carregado as configurações iniciais do ficheiro **spa942.cfg**, o próximo passo é, procurar no servidor TFTP o ficheiro único, apenas correspondente ao telefone em questão, isto é, o ficheiro necessário será aquele cujo nome é também constituído pelo endereço MAC, **spaMAC.cfg**. Portanto, todas as configurações que não são susceptíveis de sofrer alterações, e que são iguais para todos os terminais, deverão ficar no primeiro ficheiro, todas as outras ficarão no ficheiro pertencente ao próprio terminal, isto é, **spaMAC.cfg**. Na figura 3.23, pode-se visualizar o conteúdo do ficheiro inicial de configuração, utilizado em laboratório.

```

spa942.cfg
<flat-profile>
<Profile_Rule ua="na">tftp://192.168.0.100/SPA942_generico.xml</Profile_Rule>
<Resync_Periodic ua="na">5</Resync_Periodic>
<Upgrade_Enable group="Provisioning/Firmware_Upgrade">Yes</Upgrade_Enable>
<Upgrade_Error_Retry_Delay group="Provisioning/Firmware_Upgrade">120</Upgrade_Error_Retry_Delay>
<Upgrade_Rule group="Provisioning/Firmware_Upgrade">tftp://192.168.0.100/SPA$PSN-6-1-
3a.bin</Upgrade_Rule>
</flat-profile>

```

Figura 3.23 - Ficheiro inicial de configuração

O código XML acima é então o primeiro ficheiro a ser carregado pelo terminal IP durante a fase de arranque. Uma das primeiras instruções indicadas no **spa942.cfg** é o ficheiro genérico de configuração, onde contém instruções comuns a todos os telefones, como por exemplo: fuso horário, imagem de logotipo, configuração predefinidas de teclas de atalho, entre outras.

A segunda marcação (*tag*), a destacar, é a indicação de actualização do *firmware*, que neste caso está activo. Caso se queira desactivar basta substituir o **Yes** por **No** e, desta forma, o terminal IP não verifica se existe actualização disponível no servidor TFTP.

As últimas instruções a realçar é a indicação do nome do *firmware* a enviar para o telefone, caso seja para actualizar.

Sobre o ficheiro único de cada terminal, **spaMAC.xml**, este contém configurações, tais como o servidor IPPBX no qual o telefone se deve registar, o nome de apresentação

no visor, a extensão e credenciais para o registo. As configurações podem-se verificar na figura 3.24.

```

spaMAC.xml
<flat-profile>
<Proxy_1_group="Ext_1/Proxy_and_Registration">Endereço_IP_do_IPPBX</Proxy_1_>
<Display_Name_1_group="Ext_1/Subscriber_Information">Ze Nando</Display_Name_1_>
<User_ID_1_group="Ext_1/Subscriber_Information">500</User_ID_1_>
<Password_1_group="Ext_1/Subscriber_Information">123456</Password_1_>
<Auth_ID_1_group="Ext_1/Subscriber_Information">500</Auth_ID_1_>

```

Figura 3.24 - Configurações individuais de cada telefone

3.3.1 Personalização do toque

O provisionamento do toque de chamada no Linksys 942 tem que obedecer a algumas regras, aliás como todos os outros telefones IP.

Quanto às características do ficheiro de toque, devem ser, 16 bit mono, 8000 amostras por segundo e, menos que 6s de duração. É possível converter ficheiros de música no formato WAV para serem utilizados como toque no telefone IP, através da aplicação SPA *Ringtone utility*, disponibilizada no sitio *web* do fabricante¹.

O envio do toque para o telefone é feito através do servidor TFTP, como exemplifica a figura 3.25.

```

spa942_generico.cfg
<Ring1_group="Phone/Ring_Tone">n=Toque01;w=tftp://192.168.0.100/linksys_ringin.srt;c=0</Ring1>

```

Figura 3.25 - Personalização do toque de chamada

Na figura 3.25 apresenta-se a *tag* XML da configuração do toque no Linksys 942, onde o parâmetro **n** é o nome do toque, **w** corresponde ao caminho da localização do toque (pode ser HTTP ao invés de TFTP), e **c** a cadência² do áudio.

¹<https://supportforums.cisco.com/docs/DOC-9944/>

²Cadência ou *Framerate* é o número de tramas ou *frames* que o telefone vai reproduzir por unidade de tempo.

O nome da *tag* **Ring1** é a linha a que o toque será associado, neste caso é a primeira linha.

Os toques podem ser carregados a partir do ficheiro de configuração genérico, ou seja, comum a todos os telefones, ou a partir do ficheiro de configuração individual, que é identificado com o endereço MAC. No caso de utilizada a segunda forma, esta fará mais sentido se for para utilizar toques diferentes entre telefones.

3.3.2 Personalização da imagem/logotipo

O telefone IP Linksys SPA942 suporta imagens no formato BMP a preto e branco (1 bit ou monocromática) com 128x48 pixels em termos máximos, sendo recomendados 95x39 pixels.

A imagem já no formato devido deverá ser copiada para o servidor de TFTP, preferencialmente para um sub-directório **linksys**, resultando no endereço **tftp://192.168.0.100/linksys/spa942_logo01.bmp**, a ser utilizado no código XML, conforme apresentado na figura 3.26. Caso se pretenda usar HTTP em vez de TFTP é só refazer o endereço de ligação à imagem, **http://192.168.0.100/web_linksys/spa942_logo01.bmp**. É de realçar que o ficheiro da imagem de logotipo tem que ficar num directório do servidor *web*.

```
spa942_generico.cfg
(...)
<BMP_Picture_Download_URL
group="Phone/General">tftp://192.168.0.100/linksys/spa942_logo01.bmp</BMP_Picture_Download_URL>
```

Figura 3.26 - Personalização da imagem de logotipo

Este telefone tem a particularidade de só carregar definições quando estas forem diferentes, logo, quando se trata de uma imagem, esta só é descarregada do servidor se o ficheiro for diferente do que está actualmente inserido.

Neste terminal é ainda possível personalizar uma imagem de *screensaver* e um logotipo de arranque, no formato de texto que vai surgir quando o telefone for ligado ou reiniciado.

As características da imagem de *screensaver* são: 128x64 pixels; um bit por pixel, ou seja, duas cores que correspondem ao preto e branco (monocromático). Esta imagem aparece quando o telefone estiver inactivo após determinado tempo, que será especificado no código XML.

3.3.3 Personalização do idioma

O telefone IP Linksys SPA942 vem de fábrica com o idioma inglês activo por omissão. Apesar de ser possível a cada utilizador alterar no próprio telefone o seu idioma, a personalização desta opção continua necessária porque há utilizadores que poderão sentir dificuldade em fazê-lo e, desta forma, também se consegue homogeneidade em todos os terminais.

```
spa942_generico.cfg
(...)
<Dictionary_Server_Script ua="na">
serv=tftp://192.168.1.119/;d0=Português;x0=pt.xml;d1=English;x1=en.xml;d2=Catalão;x2=ct.xml
</Dictionary_Server_Script>
<Language_Selection ua="na"> Português </Language_Selection>
```

Figura 3.27 - Personalização do idioma

No código XML de personalização do idioma, pode-se destacar a configuração do parâmetro **serv** que indica o servidor onde está o ficheiro de idioma pretendido. Neste caso será utilizado um servidor TFTP. Em caso de ser empregue um servidor HTTP apenas será necessário substituir TFTP por HTTP e completar o restante endereço até ao ficheiro do idioma. Nos parâmetros **d0** e **x0** indicam-se a descrição do idioma e o ficheiro que contém o idioma em si. Como se pode verificar, ainda no mesmo código, é possível disponibilizar ao utilizador outros idiomas.

A linha final do código XML, `Language_Selection` serve para seleccionar o idioma por omissão, sendo neste caso o português.

3.3.4 Personalização da agenda

Nos manuais do SPA942, o fabricante chama à função agenda, *Personal Directory* (Directório Pessoal). Os métodos descritos para adicionar contactos são via menu do telefone ou via navegador *web* (através do seu endereço IP).

A figura 3.28 mostra o navegador *web* onde é possível adicionar contactos, com a seguinte sintaxe: `n=Ze Nando;p=709001`, onde `n` é o nome e `p` o número de telefone. Opcionalmente pode ainda ser adicionado outro parâmetro que associa um toque ao contacto, da seguinte forma: `n=Ze Nando;p=709001;r=1`, associando o toque número um ao contacto Ze Nando.

Este telefone tem a capacidade de guardar até 100 contactos na memória.



Figura 3.28 - Agenda pessoal

Este dois métodos não servem caso se pretenda ter uma agenda comum a todos os telefones. Como o fabricante não disponibiliza uma forma de carregar para o telefone uma agenda, usou-se um método descrito por Geoff Jacobs no seu blog [Jacobs07]. Jacobs utiliza o comando `wget`¹ para enviar (post) os contactos para o telefone, como mostra o exemplo da figura 3.29.

¹é um comando muito usado em Linux para descarregar ficheiros da Internet via HTTP e FTP, <http://www.gnu.org/software/wget>.

```
# Comando para enviar um contacto:
wget --post-data '28462=n%3DZe%20Nando;p%3D7090001' http://192.168.1.1/pdir.spa

# Comando para enviar dois contactos:
wget --post-data '28462=n%3DZe%20Nando;p%3D7090001&22590=n%3DSoraia%20Sa;p%3D7090002'
http://192.168.9.14/pdir.spa
```

Figura 3.29 - Comando para enviar contactos para o SPA942

Fazendo a análise ao primeiro comando da figura 3.29, é o `wget --post-data` que faz enviar via HTTP o conteúdo, que está entre plicas, para o endereço `http://192.168.1.1/pdir.spa`. No conteúdo que é para enviar, pode-se destacar o primeiro número de cinco algarismos (28462) que identifica o índice ao qual vai ser associado o contacto. Para conhecer estes índices foi necessário abrir o código HTML da página web que permite adicionar contactos (ver figura 3.28). Este código fonte pode ser visto com qualquer navegador web clicando com o botão direito do rato e depois escolher a opção “Código Fonte” ou “Ver Origem” (depende do navegador), cujo resultado pode ser visto na figura 3.30.

```
264 <tr bgcolor=#d3d3d3><td>1.<td>&nbsp;<input class="inputc"
    class="inputc" size="40" name="28334" value="" maxlength=;
265 <tr bgcolor=#dcdcdc><td>3.<td>&nbsp;<input class="inputw"
    name="28462" value="" maxlength=2047>
266 <tr bgcolor=#d3d3d3><td>5.<td>&nbsp;<input class="inputc"
    name="28590" value="" maxlength=2047>
267 <tr bgcolor=#dcdcdc><td>7.<td>&nbsp;<input class="inputw"
    name="27694" value="" maxlength=2047>
268 <tr bgcolor=#d3d3d3><td>9.<td>&nbsp;<input class="inputc"
    name="27822" value="" maxlength=2047>
269 <tr bgcolor=#dcdcdc><td>11.<td>&nbsp;<input class="inputw"
    size="40" name="27950" value="" maxlength=2047>
270 <tr bgcolor=#d3d3d3><td>13.<td>&nbsp;<input class="inputc"
    size="40" name="28078" value="" maxlength=2047>
271 <tr bgcolor=#dcdcdc><td>15.<td>&nbsp;<input class="inputw"
```

Figura 3.30 - Código fonte

Note-se que estes índices podem ser diferentes com alterações de *firmware*. No anexo B são apresentados todos índices para o Linksys SPA942 com o *firmware* 5.1.15(a).

Analisou-se também uma ferramenta SPA 942 Personal Directory Manager [Paulie10] que permite, em sistemas operativos Windows, enviar uma lista de contactos para um telefone SPA942. Na figura 3.31 é possível verificar as funcionalidades desta ferramenta. Para além de se poder adicionar o nome, número e

toque, através das células, também é possível guardar essa lista para utilizar mais tarde, bem como enviá-las para o endereço IP do telefone.

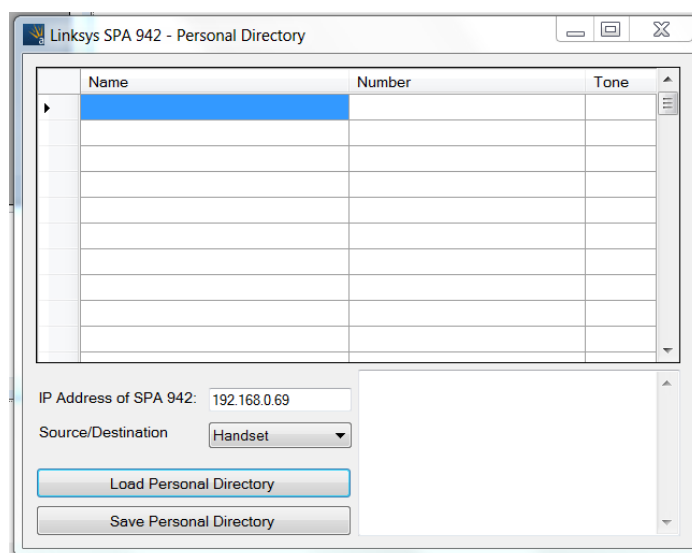


Figura 3.31 - SPA 942 Personal Directory Manager

3.3.5 Personalização das teclas de marcação rápida

As teclas de marcação rápida permitem ligar para um número já previamente estabelecido apenas pressionando um único botão continuamente. O telefone Linksys SPA 942 vem equipado com oito teclas que podem ser configuradas para marcação rápida, como se pode verificar na figura 3.32. Este conceito é semelhante às teclas de atalho de outros dispositivos, com a diferença que, aqui, para a função ser executada é necessário pressionar continuamente a tecla, desta forma, a tecla mantém a função que tinha quando pressionada só uma vez, não continuamente.



Figura 3.32 - Teclas de marcação rápida

As instruções no ficheiro XML spaMAC.cfg, para personalizar as teclas de marcação rápida, são apresentadas na figura 3.33, onde se verifica que a tecla 2 já tem configurada uma marcação rápida para o número “7090001”.

```
<Speed_Dial_2 group="User/Speed_Dial">7090001</Speed_Dial_2>
<Speed_Dial_3 group="User/Speed_Dial"/>
<Speed_Dial_4 group="User/Speed_Dial"/>
<Speed_Dial_5 group="User/Speed_Dial"/>
<Speed_Dial_6 group="User/Speed_Dial"/>
<Speed_Dial_7 group="User/Speed_Dial"/>
<Speed_Dial_8 group="User/Speed_Dial"/>
<Speed_Dial_9 group="User/Speed_Dial"/>
```

Figura 3.33 - Configuração das teclas de marcação rápida

3.4 Grandstream BT200

O Grandstream BT200 é um modelo de telefone IP existente no parque VoIP do IPL com características muito limitadas quando comparado com os restantes. Não suporta funcionalidades como agenda, imagem de logotipo e dicionário (idiomas).

Suporta apenas uma linha de comunicação SIP e suporta provisionamento, simplesmente através HTTP ou TFTP. Este telefone IP pode ser visualizado na figura 3.34.



Figura 3.34 - Telefone IP Grandstream BT200

Por omissão, este modelo de telefone IP, vem de fabrica configurado para ser provisionado através da interface *web*. É possível alterar esta definição para ficar preparado para provisionamento por TFTP com o utilitário GSutil¹. O funcionamento deste utilitário passa por estabelecer uma ligação HTTP com o telefone Grandstream e, efectuar algumas configurações especificadas através de um ficheiro de texto.

Pode-se alterar essa definição com o comando apresentado na figura 3.35, onde os vários argumentos correspondem a:

- SENHA – Palavra-passe de administrador do telefone, que por omissão é “admin”.
- IP – Endereço IP do dispositivo.
- definicoes_iniciais.cfg – Contém as definições a alterar no telefone Grandstream.

```
./gsutil -rbn <SENHA> <IP> < definicoes_iniciais.cfg
```

Figura 3.35 - Definições iniciais

Portanto, a execução deste comando será das primeiras tarefas a efectuar quando se pretende provisionar, de forma centralizada, um Grandstream BT200.

¹GSutil – É um utilitário desenvolvido por Charles Howes em Perl, que permite alterar definições em telefones IP Grandstream e pode ser descarregado de <http://www.pkts.ca/gutil.shtml>.

Na figura 3.36 apresenta-se o ficheiro **definicoes_iniciais.cfg**, onde se podem verificar as primeiras alterações a ter em conta.

```
definicoes_iniciais.cfg
# Actualização de firmware:
# P212 = 0 – Actualização por TFTP
# P212 = 1 – Actualização por HTTP
P212 = 0
# Endereço do servidor que contém as actualizações:
P192 = 192.168.0.100
# Endereço dos restantes ficheiros de configuração (provisionamento):
P237 = 192.168.0.100
```

Figura 3.36 - Conteúdo do ficheiro de definições iniciais

A ferramenta GSutil depois de descarregada¹ deve ser descompactada em **/tftp/gsutl** onde devem também permanecer, todos os ficheiros que serão por si criados. Com esta ferramenta, acompanha um modelo de configuração onde se retirou as definições apresentadas na figura 3.36 – **gxp2000.template.cfg**.

De seguida, apresentam-se mais alguns parâmetros que podem ser acrescentados ao ficheiro **definicoes_iniciais.cfg**.

O parâmetro **P145** se for 1 (um) prepara o terminal para receber do DHCP, o endereço do servidor de TFTP. Neste caso, ao colocar **P145=1**, deixa de ser necessário o parâmetro **P237** (ver figura 3.36), uma vez que este também indica ao telefone IP o servidor de provisionamento, ou seja, de TFTP. De qualquer forma, a possibilidade desta configuração é uma vantagem pois consegue-se definir remotamente, no terminal, o endereço do servidor de TFTP sem recorrer ao DHCP.

Para o telefone verificar regularmente actualizações define-se o parâmetro **P194=1**. O ultimo parâmetro, apresentado na figura 3.37, especifica o servidor para sincronização da hora.

¹<http://www.pkts.ca/gsutl.shtml>

```
# Opção 66 do DHCP
# 0 – Não se sobrepõe ao parâmetro P237
# 1 – Sim, sobrepõe-se ao parâmetro P237
P145 = 1
# Actualização automática
# 0 – Não
# 1 – Sim (Verifica todos os dias)
P194 = 1
# Servidor de NTP
P30 = 192.168.0.100
```

Figura 3.37 - Parâmetros adicionais do definicoes_iniciais.cfg

Ao contrário dos modelos de telefones VoIP já apresentados, a estrutura dos ficheiros de configuração deste modelo não é baseado em XML, como já se pôde verificar mais acima. A estrutura é baseada em parâmetros numerados em que cada um corresponde a uma função. No anexo D são apresentados todos os parâmetros de configuração deste modelo de telefone.

Para registar este telefone IP num IPPBX é necessário criar um ficheiro de configuração, semelhante ao apresentado na figura 3.38. Neste ficheiro é possível verificar a configuração do servidor onde se irá efectuar o registo da conta SIP, credenciais de autenticação e nome de apresentação no visor do telefone.

```
x500_bt200.cfg

# Nome da conta
P270 = Operadora1
# Servidor SIP
P47 = 192.168.0.100
# Proxy (servidor)
P48 = 192.168.0.100
# Utilizador SIP
P35 = 500
# Identificador de autenticação
P36 = 500
# Palavra-passe de autenticação
P34 = 500
# Nome de apresentação
P3 = Soraia Sa
# Registo SIP. 0 - no, 1 - yes
P31 = 1
```

Figura 3.38 - Configurações de registo de cada telefone

Depois de toda a configuração feita no ficheiro **x500_bt200.cfg**, é necessário compilar o ficheiro para um formato binário pois o Grandstream BT200 não reconhece ficheiros de texto. A conversão terá de ser efectuada com a ferramenta *Grandstream configuration tool*¹, disponibilizada pelo próprio fabricante tanto para sistemas operativos Windows como Unix/Linux.

A seguir apresenta-se o comando em Linux utilizado para construir o ficheiro binário de configuração através do *Grandstream Configuration Tool*. No comando, o argumento **%MAC%** corresponde ao endereço MAC do dispositivo IP, **x500_bt200.cfg** é o ficheiro de texto a ser compilado e **cfg%MAC%** será o ficheiro binário que o terminal IP irá carregar.

```
/GS_CFG_GEN/bin/encode.sh %MAC% x500_bt200.cfg cfg%MAC%
```

Figura 3.39 - Comando para criar o ficheiro de configuração do telefone

Depois de criado o ficheiro binário, este deve ser colocado no directório do servidor de TFTP, isto é, em **/tftpboot**. Pode-se utilizar o comando seguinte para este efeito.

¹ *Grandstream configuration tool*, é um utilitário para converter ficheiros de configuração no modo texto para binário, <http://www.grandstream.com/support/configurationtool.html>.

```
cp cfg000000000000 /tftpboot/
```

Figura 3.40 - Comando para transferir o ficheiro de configuração para o TFTP

No anexo A é apresentado o mesmo processo mas através da aplicação em Windows.

3.4.1 Personalização do toque

O Grandstream BT200, suporta como toque de chamada, ficheiros áudio de 16 bits com frequência máxima de 8kHz. Estes, têm de estar no formato WAV e à semelhança dos ficheiros de configurações também têm de ser convertidos para binário.

Mais uma vez, o fabricante disponibiliza um utilitário para efectuar esta conversão, tanto para sistemas operativos Windows como Unix/Linux – *Ringtone generator*¹. A seguir, apresenta-se o comando Linux para converter um toque onde, **toque.wav** é o som/música a converter e **ring1.ring** é o resultado da conversão e, tem de obrigatoriamente ter a extensão **.ring**. [GRT10]

```
./sox toque.wav ring1.ring
```

Figura 3.41 - Conversão do ficheiro audio para toque de chamada

De qualquer forma, no final da conversão é preciso renomear o **ring1.ring** para **ring1.bin** e move-lo para o directório do servidor TFTP, como se pode visualizar a seguir.

```
mv toque.ring /tftpboot/ring1.bin
```

Figura 3.42 - Comando para transferir o ficheiro para o TFTP

¹*Ringtone generator*, é um utilitário para converter ficheiros Wav para binário, <http://www.grandstream.com/ringtone.html>

Este modelo suporta três toques e terá que ter os nomes **ring1.bin**, **ring2.bin** e **ring3.bin**.

No anexo A apresenta-se este procedimento realizado com a ferramenta *Ringtone generator* mas, para sistemas operativos Windows.

3.5 X-lite

O telefone virtual X-lite não suporta nativamente provisionamento centralizado à semelhança dos *hardphones*. De qualquer forma, pode-se optar por uma forma de provisionamento muito básica, que passa por disponibilizar o X-lite numa página *web*, bem como um manual de instruções que o utilizador pode seguir até ter o *softphone* a funcionar.

Outra forma mais elaborada de provisionamento, que já se aproxima mais do que se pretende, ou seja, provisionamento centralizado, é utilizar também uma página *web* para descarregar o X-lite, mas desta vez, seria solicitado ao utilizador que se autenticasse e, a partir daí obteria acesso a um X-lite já personalizado e configurado consoante a extensão que lhe está atribuída.

A terceira proposta aqui apresentada para provisionamento do *softphone* X-lite, será a que mais se adequa ao pretendido pelo Instituto Politécnico de Leiria. A ideia é disponibilizar a instalação do X-lite numa página *web*, o utilizador só tem de o descarregar e executar o ficheiro de instalação. Durante este processo, só será necessário preencher campos básicos como o seu nome, conta de utilizador, extensão telefónica e palavra-passe. Para isto ser possível só com uma outra aplicação – o AutoIT¹.

O AutoIT é uma ferramenta livre que automatiza tarefas num sistema operativo Windows. O seu modo de funcionamento é baseado num *script* que executa comandos. Uma das formas de construir estes *scripts* é a através de cliques do rato,

¹www.autoitscript.com/

isto é, o Autoit captura os cliques que faz com o rato convertendo-os em comandos. [Brand05]

Desta forma, para criar uma instalação personalizada do X-lite, executou-se o mesmo com o AutoIT já previamente a correr. A partir daqui com uma função do “Finder Tool” da ferramenta de *scripts*, é só ir respondendo à perguntas (“Sim”, “Não”, “Seguinte”, etc) de instalação do X-lite. Desta forma a ferramenta Autoit gerou um *script* baseado nestes cliques e nas respostas. De seguida compila-se o *script* com um utilitário do AutoIT.

Por fim, tem-se um pacote do X-lite muito mais amigável para o utilizador. É só correr o ficheiro compilado que já traz todas as perguntas respondidas e configurações, sendo necessário apenas introduzir a informação de cada utilizador (nome, extensão e palavra-passe). No anexo C está apresentado detalhadamente este procedimento.

3.5.1 Síntese

Neste capítulo foi apresentado um estudo realizado a todos os telefones IP existentes no IPL, que se centrou nos processos de provisionamento de cada um. Para este estudo contribuíram a consulta aos Guias do Administrador, Manuais do Utilizador e *datasheets* dos equipamentos, análise de ficheiros de configuração e testes realizados em laboratório.

Idealizou-se e apresentou-se ainda uma forma de se conseguir provisionamento de um *softphone*, já que o seu uso se prevê ser muito pretendido pela comunidade.

Nos telefones IP, conseguiu-se também ficar com uma noção das características e potencialidades de cada um, onde se pode destacar pela negativa o Grandstream BT200, devido às suas funcionalidades muito limitadas quando comparadas com os restantes aqui estudados.

4 Arquitectura

Neste capítulo irá apresentar-se uma arquitectura para a solução requerida na presente dissertação – Provisionamento em Infraestruturas de Voz Sobre IP. Esta arquitectura, que não é mais do que um conjunto de processos representados em esquemas do tipo fluxograma, foi desenvolvida tendo em conta o estudo efectuado e apresentado no capítulo 3.

Embora os processos práticos de provisionamento para cada terminal seja diferente, a arquitectura será apresentada de forma genérica, ou seja, será descrito um conjunto de processos que se aplicarão a todos os terminais.

Na primeira secção são apresentados os requisitos a ter em conta para o desenvolvimento da plataforma. De seguida é apresentado o modelo de arquitectura geral da plataforma e abordado as fases que um terminal IP percorre, até ser provisionado. As restantes secções descrevem os processos de provisionamento necessários, para atingir os objectivos definidos na tabela 4.1.

4.1 Requisitos

Com a finalidade de se apresentar uma estrutura, adequada às necessidades do Instituto Politécnico de Leiria, nesta secção serão apresentados os requisitos mínimos

exigidos à solução de provisionamento, ou seja, funcionalidades ou objectivos a atingir com o modelo de provisionamento. Estes requisitos resultaram de uma reunião de trabalho com a Unidade de Administração de Redes e Sistemas (UARS) do IPL. Nessa reunião, definiram-se os requisitos ou funcionalidades, que estão sintetizadas na tabela 4.1, onde é indicado o requisito em questão e a descrição do mesmo.

Requisito	Descrição
Configuração automática do telefone	Atribuir um IP, extensão, Nome no Visor, entre outros, a um telefone IP.
Actualização de <i>firmware</i>	Possibilidade de actualizar o <i>firmware</i> de todos os telefones IP.
Configuração de rede	Atribuição de IP a um telefone IP novo, ou alteração de um já existente.
Configuração da linha	Configuração de uma conta SIP.
Palavras-passe	Atribuição de senhas aos terminais VoIP, de forma a proteger as suas configurações dos utilizadores
Reiniciar os dispositivos	Possibilidade de reiniciar os equipamentos provisionados.
Personalizar toques e logos	Capacidade de alteração de um toque de um telefone ou do logo do visor.
Visualização de agenda nos telefones	Possibilidade do telefone IP ter uma agenda no visor, sincronizada com a AD.
Plataforma <i>web</i>	Haver possibilidade das funcionalidades serem executadas através de uma consola <i>web</i> .
Plataforma independente	A plataforma a propor terá de funcionar de forma independente à arquitectura VoIP já em funcionamento.

Tabela 4.1 - Requisitos para solução de provisionamento

A independência da plataforma é um requisito importante a ter sempre em conta pois pretende-se que ao colocar em actividade a presente plataforma, não seja necessário efectuar grandes alterações nos servidores já em produção.

4.2 Modelo de arquitectura

Após o estudo efectuado aos dispositivos VoIP, desenvolveu-se um modelo de arquitectura que descreve a interacção desde o telefone VoIP até à interface *web*.

Este modelo foi idealizado tendo por base o provisionamento do dispositivos VoIP através de mecanismos disponibilizados na interface *web*.

Na figura 4.1, está apresentada a arquitectura de comunicação entre o dispositivo VoIP e a plataforma de provisionamento – Modelo de arquitectura.

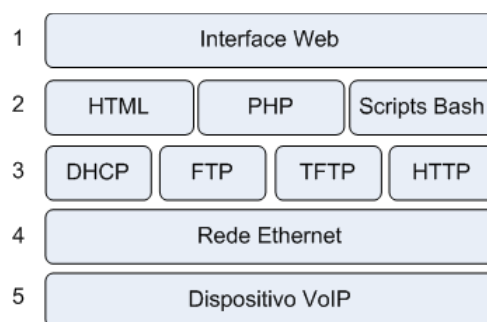


Figura 4.1 - Modelo de arquitectura

Analisando a figura 4.1, que representa o modelo de arquitectura apresentado na presente dissertação, tem-se na primeira camada a componente gráfica da plataforma – Interface *web*. Na segunda camada, encontram-se representadas as tecnologias de programação que foram utilizadas como ponte entre a interface *web* e a terceira camada – Serviços de rede. Na prática esta segunda camada faz executar as ordens da Interface *web*, de forma a alterar as configurações necessárias nos serviços de rede para se atingir o provisionamento do terminal VoIP. Na camada dos serviços de rede, o DHCP é responsável pela atribuição das configurações de rede ao terminal IP, quando este se liga à infra-estrutura *Ethernet*. Os restantes três serviços, FTP, TFTP e HTTP, são necessários para providenciar o provisionamento dos parâmetros e configurações dos terminais VoIP. A quarta camada – Rede *Ethernet*, é a camada de mais baixo nível desta arquitectura, que engloba a rede física e os protocolos do

modelo TCP/IP. Em último, está o dispositivo VoIP que é o alvo de toda a plataforma de provisionamento.

De seguida, a secção 4.3 explicará mais detalhadamente o que acontece entre a camada 5 e a camada 3. As secções 4.4, 4.5, 4.6, 4.7 e 4.9, explicam o funcionamento interno da camada 1 – Interface *web*.

4.3 Fases de provisionamento

O primeiro contacto do terminal IP com um sistema de provisionamento surge através de um pedido ao servidor DHCP. Nesta primeira fase, em que o terminal se liga à rede e requer um endereço IP, o serviço de DHCP é indispensável, pois é ele que além de atribuir todas as configurações de rede mais usuais, como o endereço IP, a porta de ligação e o DNS, também vai indicar ao terminal qual é o servidor que contém as suas definições ou a imagem de arranque. Um serviço de DHCP funciona por intermédio de opções, onde cada uma corresponde a uma configuração a atribuir. Conforme abordado na secção 2.4.4, a opção 66 indica o servidor TFTP para a rede em questão. A opção 67, necessária a alguns terminais, indica qual a imagem de arranque que o mesmo deve carregar.

Na figura 4.2 é ilustrado um esquema detalhado das fases de provisionamento de um terminal VoIP, desde que este se liga à rede até se encontrar registado num IPPBX.

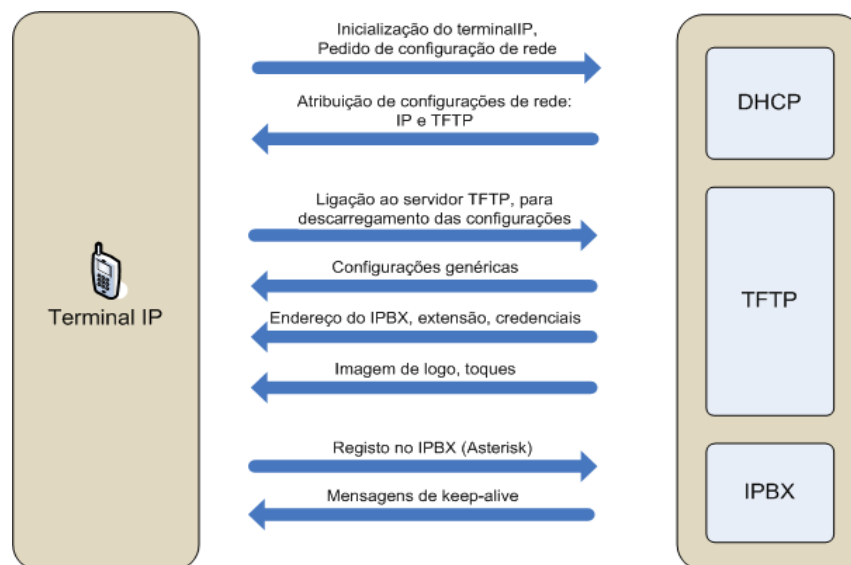


Figura 4.2 - Fases de arranque de um terminal IP

Como pode ser verificado na figura acima, o arranque típico de um terminal IP pode ser distinguido em três etapas, aí representadas. A primeira etapa, representa o arranque inicial do dispositivo, onde são atribuídas todas as condições para a ligação à rede e por sua vez ao servidor de TFTP. Na etapa seguinte, o terminal conecta-se ao servidor TFTP onde descarrega todos os ficheiros necessários ao seu funcionamento. Por fim, o telefone ou dispositivo IP autentica-se no IPPBX com as informações de registo e credenciais recebidas na etapa anterior.

É de salientar que este processo só terá êxito caso já existam as configurações necessárias para os dispositivos VoIP carregarem todas as definições de provisionamento. Estas instruções são as directivas no ficheiro de configuração do serviço de DHCP. Neste processo devem existir os ficheiros característicos de cada modelo com todos os parâmetros devidamente atribuídos (endereço do IPPBX, extensão, credenciais, toques, imagem de logotipo, idioma, etc).

Nos sub-capítulos seguintes serão descritos procedimentos a efectuar antes de ligar determinado telefone IP à rede. Serão também apresentados métodos e mecanismos indispensáveis a uma plataforma de provisionamento.

4.4 Provisionamento de um novo terminal

A configuração de um novo terminal a provisionar obedece a algumas regras e processos normalmente efectuados em ficheiros de texto. Mais uma vez, aqui pretende-se generalizar o processo, não individualizando uma marca ou modelo de terminal telefónico IP. As particularidades de provisionamento dos diferentes modelos de terminais IP existentes no IPL, já foram abordadas no sub-capítulo 2.2.

Depois do arranque do terminal, este aguarda que sejam disponibilizados ficheiros de configuração no servidor TFTP, FTP ou mesmo HTTP. Normalmente, todos necessitam de pelo menos um ficheiro composto pelo seu endereço MAC. Este endereço é utilizado no nome do ficheiro por ser único, não havendo a possibilidade de repetição.

Quando se pretende provisionar um novo terminal, é necessário preparar preparado cada ficheiro de configuração que lhe é intrínseco. O processo é todo descrito no diagrama da figura 4.3. O diagrama descreve as tarefas a realizar para se começar a provisionar um dispositivo VoIP. Verifica-se que é preciso criar ficheiros de configuração e dentro dos mesmos alterar vários parâmetros. Na configuração do serviço de DHCP, também é inevitável adicionar uma directiva para o endereço MAC do terminal IP em questão.

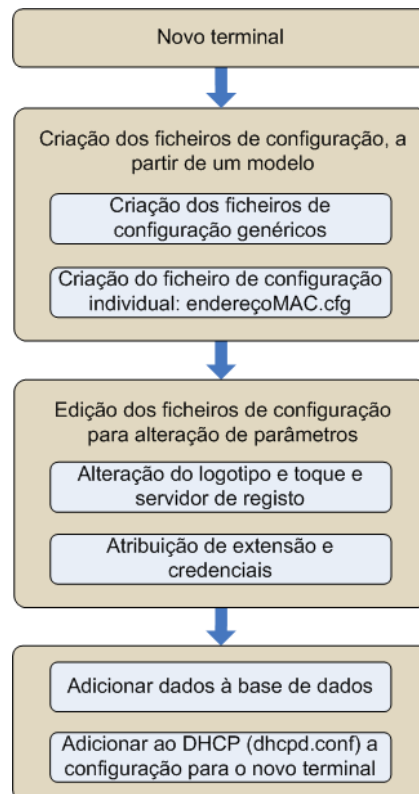


Figura 4.3 - Provisionar um novo terminal

4.5 Terminais registados no Asterisk

Uma característica importante que a plataforma deve conter, é ser totalmente independente da estrutura VoIP já existente, ou seja, funcionar num servidor à parte, conforme já descrito na secção 4.1.

A plataforma terá de identificar em determinados momentos as extensões que estão registadas e se estão ligadas. Desta forma, também se conseguirá verificar se um telefone acabado de provisionar se conseguiu registar com sucesso no IPPBX da instituição.

Tornou-se pois necessário desenhar uma arquitectura que descrevesse a interoperabilidade entre a plataforma de provisionamento e o sistema VoIP,

actualmente instalado no Instituto Politécnico de Leiria. Esta arquitectura efectivou-se através de uma rotina de instruções executadas periodicamente, no servidor Asterisk onde se registam os utilizadores. Este procedimento cria um ficheiro com todos os utilizadores (extensões) registados. O ficheiro poderá ser criado recorrendo a um comando da consola de administração do Asterisk, que identificará quais as extensões que estão, ou não, activas. Posteriormente, este ficheiro será copiado para a plataforma de provisionamento ou ficará disponibilizado por HTTP.

O primeiro método idealizado teve por base a cópia da informação através do comando SCP (*Security Copy Protocol*). O SCP é um comando existente em sistemas operativos Unix ou derivados, que permite copiar ficheiros pela rede, de forma segura, sendo toda operação é feita sobre SSL (Secure Socket Layer) [Dierks99].

O diagrama da figura 4.4 representa uma rotina que deve ser executada no servidor Asterisk. Este servidor serve de IPPBX, ou seja, inclui o registo dos telefones IP. Analisando a figura, verifica-se que primeiro é gerado um ficheiro com as extensões ou terminais registados, copiado posteriormente para a plataforma de provisionamento. O passo seguinte consta do tratamento do ficheiro, para que fique num formato que se consiga inserir na base de dados da plataforma.

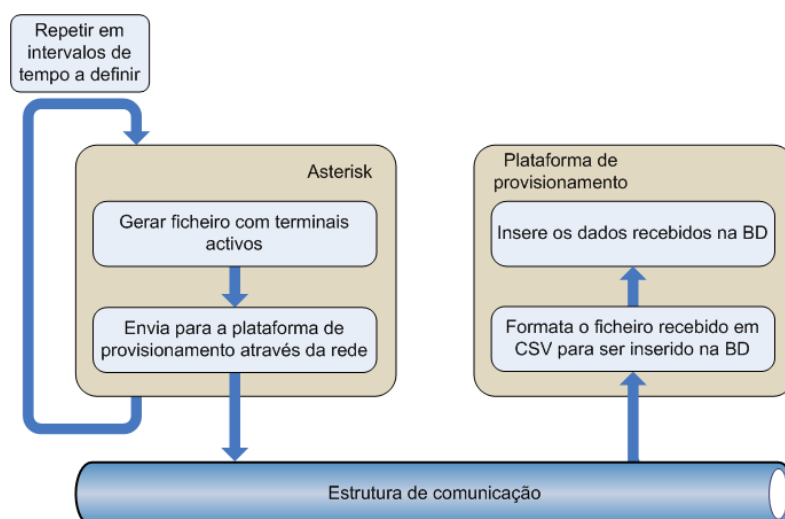


Figura 4.4 - Diagrama de interoperabilidade usando SCP

O segundo método também possível de implementar, baseia-se na disponibilização da informação exportada do Asterisk para uma área *web*, onde a plataforma tenha facilmente acesso via HTTP. A diferença para o primeiro método está na interoperabilidade entre os dois sistemas. A segunda forma tem a vantagem de poupar recursos ao servidor Asterisk, uma vez que este não tem de enviar informação para a plataforma. Logo, é a situação preferencial pois cumpre eficazmente o requisito de não interferir com o sistema de telefonia IP, já em produção.

Na figura 4.5 pode ser visualizada a arquitectura deste processo, redesenhado para disponibilizar a informação por HTTP, onde se pode verificar a diferença das funções desempenhadas no Asterisk.

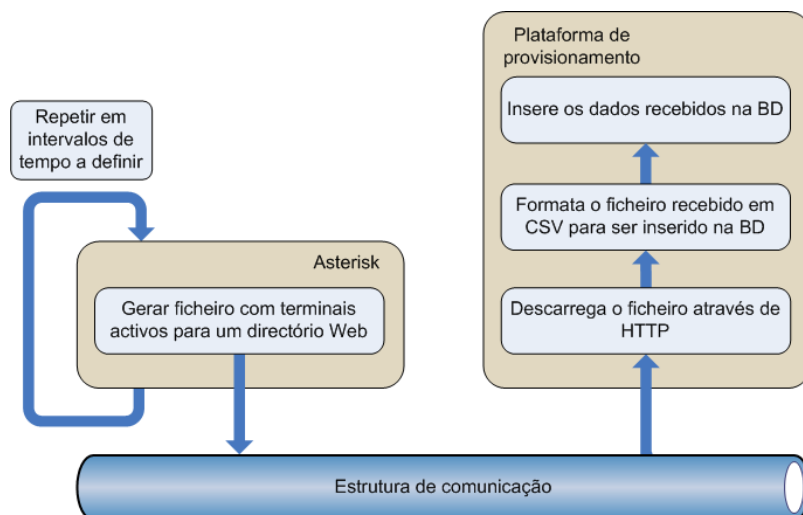


Figura 4.5 - Diagrama de interoperabilidade usando HTTP

4.6 Terminais adicionados à plataforma

A plataforma deverá ter uma funcionalidade para listar tanto os terminais registados no IPPBX como os terminais adicionados à plataforma de provisionamento. Os terminais que estejam adicionados à plataforma poderão não estar registados no

IPPBX, nem tão pouco terem extensão criada. Por esta razão, deverão existir estas duas listagens.

Uma vez que nas etapas descritas em 4.4 e 4.5, as informações sobre os terminais já são guardados numa BD, o procedimento para identificar os que estão actualmente ligados passa por efectuar uma simples consulta à BD.

O processo é semelhante para as duas listagens e está representado na figura 4.6. Pode-se verificar que a pesquisa dos dados, sobre os terminais IP, é efectuada com um comando de consulta à tabela base de dados MySQL – `select * from t_terminais`, onde `t_terminais` é a tabela que contém a informação. Os dados sobre os terminais deverão ser apresentados numa página *web*.

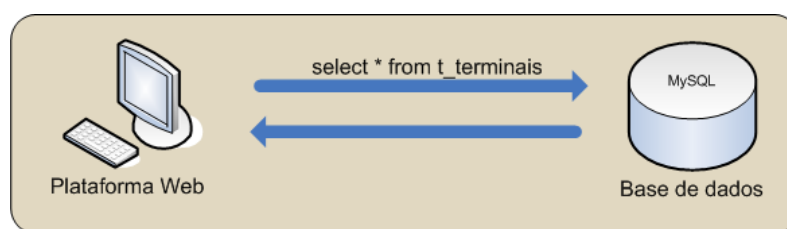


Figura 4.6 - Processo para listar terminais

4.7 Configurações em massa

Será indispensável, para o administrador dos telefones IP, alterar em massa diversos parâmetros, tais como: imagem de logotipo, toque do telefone ou mesmo o idioma. Este processo, embora simples, poderá tornar-se complexo de implementar, pois a plataforma terá que ter a capacidade de editar vários ficheiros de configuração e encontrar parâmetros, em locais específicos, para os apresentar na interface e permitir a sua alteração. Na figura 4.7 são apresentadas as fases que este processo deverá ter.

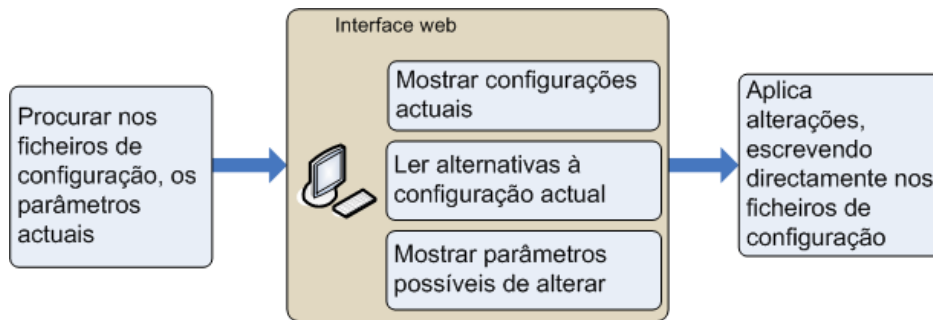


Figura 4.7 - Processo para alteração de configurações em massa

Analisando a figura 4.7, a primeira fase deste processo será ler os ficheiros de configuração e verificar os actuais parâmetros, de forma a serem mostrados na consola *web*. De seguida, a interface deverá ler as possíveis alternativas à configuração actual, por exemplo, no caso de alteração do toque de chamada, esta deve ler os ficheiros de toques disponíveis no disco da plataforma e apresenta-los ao administrador. Depois de seleccionadas as alterações, a última fase consiste na aplicação das alterações efectuadas gravando as mesmas nos ficheiros de configuração.

4.8 Sincronização dos contactos com a AD

A sincronização com a AD é uma mais valia por permitir que todos os telefones IP terão uma agenda comum e em constante actualização. É na AD que é baseada toda a autenticação de aplicações e serviços do IPL, sendo assim, esta contém actualizado todos os funcionários da instituição, com vários campos associados, tais como: Nome completo, nome de utilizador, correio electrónico, localização, extensão telefónica, entre outros.

A agenda de contactos para os terminais IP aqui estudados pode ser disponibilizados através de um ficheiro de texto formato em XML, sendo que este formato varia consoante o modelo de telefone.

O processo de sincronização descrito na figura 4.8 passa por efectuar-se uma consulta à AD na unidade organizacional (OU) que contém os funcionários. Nesta consulta deverá ser especificado os campos que se pretende ler, sendo que os essenciais para a agenda são: Nome e extensão. De seguida os dados vindos da AD são formatados e guardados num directório da plataforma. Por fim, estes ficheiros serão disponibilizados para os três modelos de telefones IP que suportam agenda (figura 4.8).

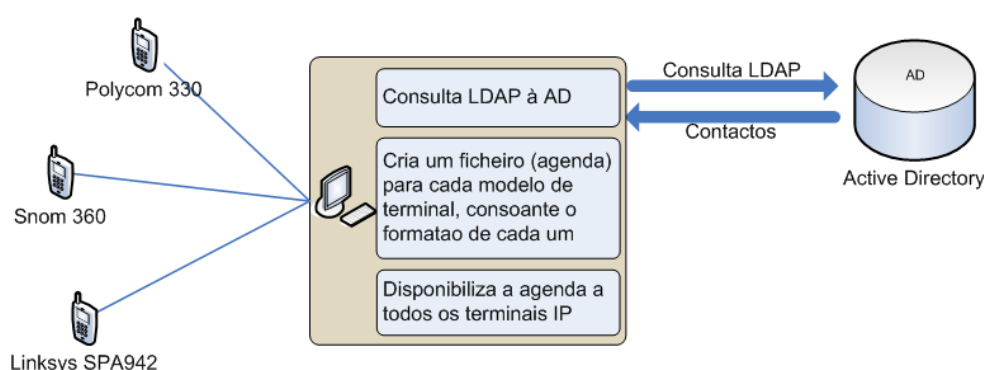


Figura 4.8 - Processo de sincronização dos contactos com a AD

4.9 Tolerância a falhas

Por ser um recurso muito utilizado numa empresa, a comunicação telefónica terá que ser eficaz, sendo que a sua falha tem de ser evitável a todo o custo. Tipicamente, uma falha nas comunicações telefónicas numa empresa ou instituição é considerada grave, uma vez que é um recurso muito utilizado no contacto com clientes e fornecedores e mesmo entre os próprios funcionários.

Para que as falhas na plataforma não se repercutam no funcionamento dos telefones, desenvolveu-se um conjunto de procedimentos, apresentados nesta secção. Vão integrar a arquitectura da plataforma de provisionamento, baseado na adição de um *cluster* para tolerância a falhas que possa garantir alta disponibilidade do serviço.

A plataforma é constituída por vários serviços, sendo o DHCP o mais crítico para o bom funcionamento dos telefones, pois se algum telefone tiver que reiniciar não

conseguirá obter configurações de rede e, como consequência disso, não se conseguirá registar no IPPBX, nem tão pouco será provisionado pela plataforma.

A arquitectura de tolerância a falhas proposta, destaca-se em duas funcionalidades, uma relacionada com a sincronização de toda a informação com outra plataforma (secundária) e, a segunda, com a monitorização da plataforma primária.

A monitorização passa por ser verificada constantemente a operacionalidade, dos serviços. Em caso de algum serviço sofrer uma interrupção, a plataforma secundária inicia os seus serviços e passa a ter o controlo sobre os dispositivos VoIP daquela rede.

A figura 4.9, mostra a abordagem de tolerância a falhas. Inicialmente, o sistema encontra-se a funcionar normalmente. Quando acontece algum problema nos serviços essenciais à plataforma, ou de conectividade, o servidor secundário inicia os mesmos serviços. Desta forma, se algum telefone reiniciar ou outro arrancar pela primeira vez, irão ser provisionados pela plataforma secundária, que contém todos os dados da primária.

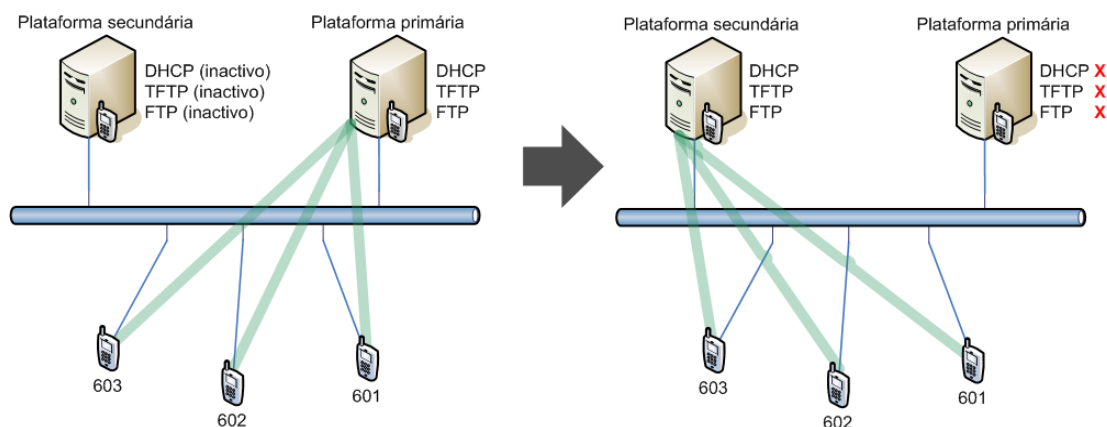


Figura 4.9 - Esquema de tolerância a falhas

Os dados a replicar entre as duas plataforma são:

- Base de dados – Informação dos terminais adicionados.

- Todo o conteúdo da pasta /TFTP – Ficheiros de configuração de todos os terminais provisionados por TFTP.
- Todo o conteúdo da pasta /home/polycom – Ficheiros de configuração dos telefones Polycom 330.

Estes dados, a manter nas duas plataformas (primária e secundária), poderão ser replicados através de *scripts* agendados periodicamente a partir do servidor primário. Estes *scripts*, conforme se pode verificar na figura 4.10, serão baseados em comandos **scp** para transferência de ficheiros e no **mysqldump**, para replicação da base de dados MySQL.

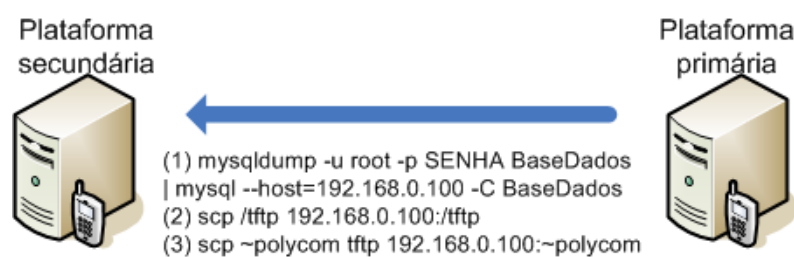


Figura 4.10 - Replicação dos dados de provisionamento

5 Protótipo

A arquitectura proposta no capítulo 4 define um conjunto de processos que servirá de base para a construção de uma plataforma de provisionamento a aplicar em ambiente real. Neste capítulo irá propor-se um protótipo que servirá por um lado para colocar em prática o estudo aqui realizado e por outro para avaliar e testar a arquitectura desenhada.

O protótipo será em formato *web* por permitir facilidade de administração e acesso em qualquer lado a partir de um simples navegador de Internet. Visto ser recomendável a opção por soluções e componentes livres, o protótipo da plataforma de provisionamento foi baseado em PHP e Mysql, estando estes assentes no sistema operativo Linux.

Serão aqui descritas as funções desenvolvidas como a preparação do cenário, idealização da interface *web* bem como todas as funcionalidades que permitem efectuar o provisionamento dos terminais IP, a partir desta consola.

Na tabela 5.1 encontram-se resumidas as funcionalidades implementadas neste protótipo.

Telefone	<ul style="list-style-type: none"> - Possibilidade de adicionar um telefones, ou vários, através do endereço MAC ou através de um leitor de código de barras. - Actualização de <i>firmware</i>. - Selecção de uma imagem de logotipo na interface <i>web</i>. - Selecção do toque de chamada pré-definido na interface <i>web</i>. - Possibilidade de escolher o ficheiro com agenda. - Possibilidade de pré-definição do idioma para todos os telefones. - Visualização dos telefones adicionados à plataforma e registados no Asterisk. - Alteração da senha de administração do telefone em massa. - Sincronização da agenda com a AD.
Softphone	<ul style="list-style-type: none"> - Disponibilização de dois <i>softphones</i> diferentes já previamente provisionados para transferência na plataforma <i>web</i>.
Extensões	<ul style="list-style-type: none"> - Visualização de detalhes de uma extensão (Modelo do telefone, endereço MAC, IP). - Reiniciar os telefones todos ou selectivamente por modelo, extensão ou IP.
Administração	<ul style="list-style-type: none"> - Alteração da senha de administração para todos os telefones.
Monitorização	<ul style="list-style-type: none"> - Janela que permite monitorização dos telefones ligados/desligados. - Janela que permite monitorização dos serviços da plataforma.
Plataforma	<ul style="list-style-type: none"> - Visualização das configurações da plataforma (IP, caminhos dos ficheiros de configuração, executáveis) - Agendar cópias de segurança. - Envio de relatório da actividade dos serviços da plataforma e dos telefones.

Tabela 5.1 - Funcionalidades desenvolvidas

5.1 Cenário

A preparação do cenário começou com a instalação de uma distribuição Linux numa máquina com o objectivo de ser o servidor de provisionamento. A distribuição escolhida foi a CentOS [Membrey09] por ser bastante focada para sistemas empresariais e por possuir grande estabilidade, no que respeita a operacionalidade e funcionamento [Turnbull09].

Depois da instalação do sistema operativo, seguiu-se a instalação das componentes fundamentais à plataforma de provisionamento, como o serviço de DHCP, servidor de TFTP, MySQL e o PHP.

Depois do servidor preparado, o cenário ficou completo com a ligação de um modelo de cada telefone existente no IPL. Na figura 5.1 pode-se observar o esquema do cenário com todos os componentes, usado em ambiente de laboratório.

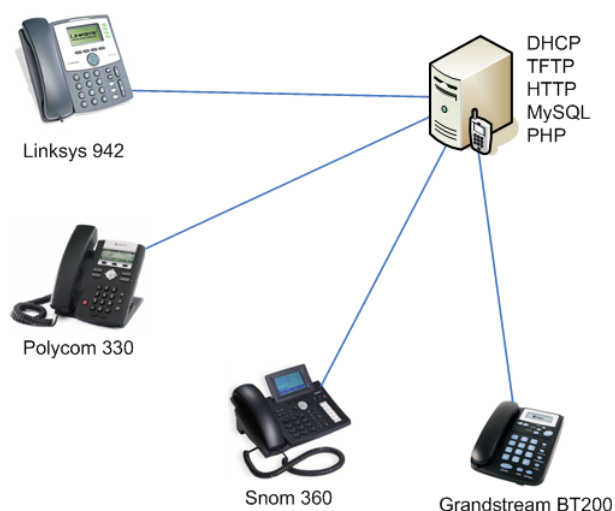


Figura 5.1 - Cenário utilizado em laboratório

Este cenário permitiu avaliar e comprovar, na prática, o provisionamento aos dispositivos representados na figura 5.1, ou seja, todas as funcionalidades apresentadas no capítulo 3 foram testadas neste cenário, bem como a arquitectura proposta no capítulo 4.

Inicialmente, efectuaram-se todos os testes de provisionamento apresentados no capítulo 3, ou seja, de forma manual recorrendo à edição dos ficheiros de configuração de cada telefone.

Como já foi referido no capítulo 3, este estudo foi baseado essencialmente em documentação disponibilizada pelos fabricantes, onde muitas vezes a informação era incompleta e pouco explícita. Este cenário foi bastante importante não só para comprovar as operações descritas nos manuais ou *admin guides* e *datasheets* dos equipamentos, mas também para descobrir certas funções de provisionamento que não eram explícitas ou que não eram mesmo relatadas. Foi o caso do Linksys SPA 942 em que o provisionamento das suas funcionalidades foi muito à base da análise dos ficheiros de configuração do dispositivo. Isto é, efectuava-se uma alteração manualmente através da página *web* do telefone e depois estudava-se no ficheiro de configuração do telefone quais tinham sido as alterações concretizadas. Este método

foi possível porque este telefone IP possibilita que o seu ficheiro de configuração seja descarregado via HTTP, através do endereço `http://IP_do_TelefoneSPA942/admin/spacfg.xml`.

O mesmo cenário serviu ainda de suporte ao desenvolvimento da plataforma *web* que automatizou as tarefas de provisionamento manuais, evitando a necessidade de se recorrer à edição de variados ficheiros de configuração.

5.2 Idealização da interface

Desenvolveu-se uma interface gráfica, destinada à administração da configuração dos telefones, organizada em quatro áreas agregando funcionalidades análogas, como se pode visualizar na figura 5.2. No grupo “Plataforma” pode-se verificar os estados dos vários serviços a correr, visualizar as configurações correntes da plataforma, programar cópias de segurança do sistema, activar e programar relatórios e enviar ficheiros para a plataforma. Em “Terminais” é possível adicionar telefones para provisionar, um ou vários de uma só vez, disponibilizar a listagem tanto de terminais adicionados a esta plataforma como também de terminais actualmente registados no servidor IPPBX, descarregar *softphones* previamente provisionados e sincronizar a agenda de contactos com a AD. O terceiro grupo serve para mostrar detalhes de uma extensão; reiniciar telefones baseados no número da extensão, IP ou modelo; ou ainda descarregar manuais do utilizador que poderão estar disponíveis na plataforma. Nas “Configurações em massa” alteram-se definições, como por exemplo *firmware*; toque de chamada; imagem de logotipo; ficheiro de agenda ou o idioma predefinido. Estas alterações afectam todos os terminais do modelo onde se efectuarem as definições.



Figura 5.2 - Organização das funcionalidades na interface *web*.

Na figura 5.3 apresenta-se o aspecto final da página *web* inicial da plataforma de provisionamento, onde são também mostrados alguns números gerais, como o número total de terminais adicionados, quantos estão a ser provisionados e quantos não estão abrangidos pelas configurações de provisionamento.

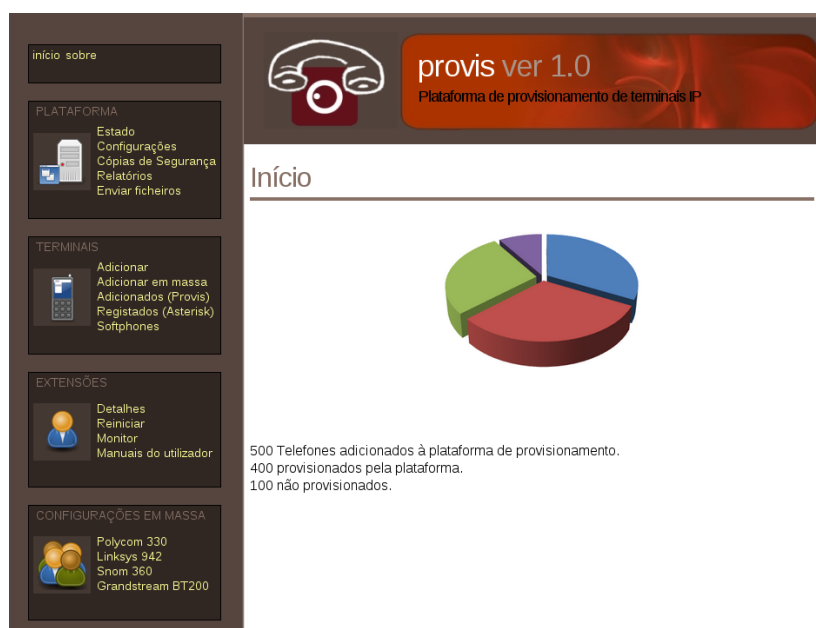


Figura 5.3 - Página inicial da plataforma de provisionamento

Nas secções seguintes será analisado detalhadamente, o desenvolvimento destes grupos de funcionalidades.

5.3 Grupo “Plataforma”

O grupo de funcionalidades “Plataforma” é uma área destinada ao funcionamento da interface *web* em geral. Aqui poderá visualizar-se o estado dos serviços essenciais ao seu funcionamento (SSH, DHCP, HTTP, TFTP, FTP e MySQL), as configurações mais importantes da plataforma, programar cópias de segurança do sistema, programar relatórios de monitorização, e enviar ficheiros para a plataforma.

5.3.1 Função “Estado”

A função “Estado” verifica o estado dos serviços mais críticos da plataforma de provisionamento e mostra na interface *web* o estado dos mesmos, que pode ser serviço “A correr” ou serviço “Parado”. Esta monitorização de serviços foi possível integrando o projecto GPL PhpWatchDog¹, na interface *web* de provisionamento. Para esta integração foi necessário alterar o código do PhpWatchDog para obter o aspecto apresentado na figura 5.4, e os serviços a monitorizar serem também os apresentados na mesma figura.



Estado		
Anfitrião : 192.168.9.244 - provis		
Porto	Serviço	Estado
22	ssh	 A correr
80	http	 A correr
69	tftp	 Parado
3306	mysql	 A correr

Figura 5.4 - Função Estado

¹<http://phpwatchdog.sourceforge.net/>

5.3.2 Função “Configurações”

Aqui é possível ver as configurações gerais da plataforma, tais como o seu endereço IP e os outros serviços tais como o TFTP ou FTP, que poderiam estar a funcionar noutra servidor. São também apresentados os caminhos de vários ficheiros de configuração e executáveis.

Estas configurações, que estão apresentadas na coluna “Configuração” da figura 5.5, estão guardadas na base de dados, são lidas pela interface *web* e guardadas em variáveis, de forma a serem utilizadas pelo código desenvolvido.

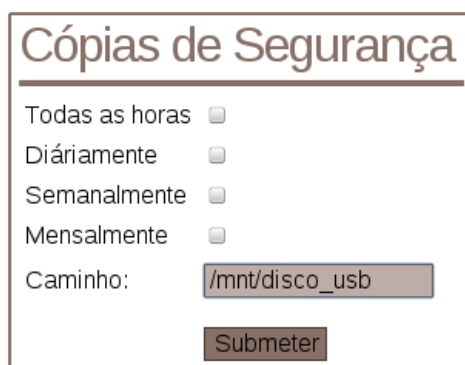
Configurações da Plataforma		
Tipo	Configuração	Afectação
IP da Plataforma	192.168.9.244	Plataforma
IP do servidor de TFTP	192.168.9.244	Plataforma
IP do servidor HTTP	192.168.9.244	Plataforma
IP do servidor de FTP	192.168.9.244	Plataforma
Ficheiro de configuração do DHCP	/etc/dhcpd.conf	Plataforma
Dir. do servidor TFTP	/tftpboot	Plataforma
Dir. dos Polycom 330	/home/polycom	Polycom 330
Dir. da plataforma	/opt/lampp/htdocs/provis	Plataforma
Executável do PHP	/opt/lampp/bin/php	Plataforma

Figura 5.5 - Configurações da plataforma

5.3.3 Função “Cópias de segurança”

Esta função serve apenas para activar cópias de segurança da plataforma. É possível realizarem-se cópias de hora a hora; todos os dias; uma vez por semana e uma vez por mês. Estas cópias permitem salvaguardar todos os dados da plataforma localizados em */tftpboot*, */home/polycom*, MySQL e interface *web*, para uma pasta que pode ser local, remota, ou um dispositivo USB, como se pode verificar na figura 5.6. No caso de ser um ponto remoto, este tem de ser previamente montado no próprio sistema operativo.

O agendamento das cópias é feito no CRON¹, ou seja, o *script* que executa as seguranças é copiado para `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` ou `/etc/cron.monthly`, conforme a *checkbox* que se selecciona.



The image shows a web form titled "Cópias de Segurança". It contains four radio button options for frequency: "Todas as horas", "Diariamente", "Semanalmente", and "Mensalmente". Below these is a text input field labeled "Caminho:" with the value "/mnt/disco_usb" entered. At the bottom of the form is a "Submeter" button.

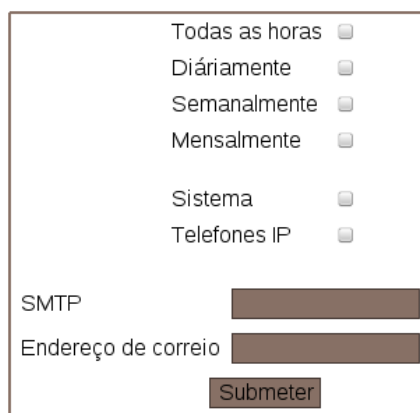
Figura 5.6 - Cópias de segurança da plataforma

5.3.4 Função “Relatórios”

Os relatórios podem ser activados com a mesma frequência das cópias de segurança, tendo como objectivo o de informar o administrador do estado do sistema. Os dados são enviados via correio electrónico e podem indicar o estado da plataforma como também o estado dos telefones IP. Aqui é altamente recomendável que a frequência de envio de alertas seja “Todas as Horas”, de forma a minorar o tempo de intervenção em caso de algum problema ocorrer.

Conforme se pode ver na figura 5.7 para além da possibilidade de seleccionar a frequência do envio dos relatórios está também disponível a opção do que se quer enviar, isto é, o relatório do sistema e o relatório dos telefones IP, e ainda a possibilidade de configurar o servidor SMTP e o endereço de correio electrónico que irá receber os alertas.

¹CRON – é o nome do mecanismo em sistemas operativos Unix que permite agendar tarefas.



Form for system reports configuration. It includes a list of frequency options: 'Todas as horas', 'Diariamente', 'Semanalmente', and 'Mensalmente', each with an unchecked checkbox. Below these are two more options: 'Sistema' and 'Telefones IP', also with unchecked checkboxes. At the bottom, there are two input fields: 'SMTP' and 'Endereço de correio', both containing a brown rectangular placeholder. A 'Submeter' button is located below the 'Endereço de correio' field.

Figura 5.7 - Relatórios do sistema

Na figura 5.8 é apresentado um exemplo de uma mensagem de correio electrónico a alertar uma falha no serviço HTTP, porta 80, do servidor provis.

```

Para: paulo.gomes@ipleiria.pt
Assunto: Alerta! > provis <
Mensagem: O serviço tftp não está a responder no endereço IP 192.168.9.244 !!!
Porto: 80
Nome do anfitrião: provis
De: admin-provis@localhost
Responder para: administrador@localhost
Versão do PHP: 5.3.1
Array
(
    [Servers] => Array
        (
            [host] => 192.168.9.244
            [server_name] => provis
            [service] => Array
                (
                    [1] => 22
                    [2] => 80
                )
        )
)

```

Figura 5.8 - Exemplo de uma mensagem de alerta

5.3.5 Função “Enviar Ficheiros”

Esta funcionalidade permite ao administrador enviar facilmente ficheiros para a plataforma. Estes ficheiros podem ser toques de chamadas, imagens de logotipo,

firmwares, ou mesmo manuais do utilizador. A interface para fazer este envio é apresentada na figura 5.9 onde se pode visualizar os vários tipos de ficheiros a enviar. Estes ficheiros serão colocados pela plataforma nas pastas apropriadas, para cada um.

A interface 'Enviar ficheiros' apresenta uma lista de cinco categorias de ficheiros a serem enviadas. Cada categoria possui um botão 'Escolher ficheiro', um status atual e um botão 'Enviar'.

Enviar ficheiros			
Toques de chamada	Escolher ficheiro	Nenhum ficheiro seleccionado	Enviar
Imagens de logotipo	Escolher ficheiro	Nenhum ficheiro seleccionado	Enviar
Agenda	Escolher ficheiro	Nenhum ficheiro seleccionado	Enviar
Firmware	Escolher ficheiro	Nenhum ficheiro seleccionado	Enviar
Manuais	Escolher ficheiro	Nenhum ficheiro seleccionado	Enviar

Figura 5.9 - Função "Enviar Ficheiros"

5.4 Grupo “Terminais”

Este grupo junta as funções relacionadas com a adição e listagem de terminais na interface *web*. É constituído por quatro funções como se pode verificar na figura 5.2.

5.4.1 Função “Adicionar”

Esta função serve para adicionar um terminal, individualmente, à plataforma solicitando vários campos, tais como: extensão, senha, endereço MAC, modelo do terminal IP, localização, e a indicação se é para provisionar pela plataforma. Caso esta última opção não seja seleccionada, o dispositivo apesar de ser adicionado à base de dados da plataforma, não será alvo de provisionamento da mesma. Desta forma, este terminal apenas gozará do serviço de DHCP da plataforma e as configurações terão de ser efectuadas manualmente. Esta opção pode servir para

adicionar telefones IP que já estejam em funcionamento e, para os quais não se pretende, alterar qualquer configuração.

Na figura 5.10 pode-se visualizar um exemplo do processo da função “Adicionar”.

A interface 'Adicionar terminais IP' apresenta um formulário com os seguintes campos e elementos:

Extensão	Senha	MAC	Modelo	Localização	Provisionar
500	12345	10:04:f2:1e:ac:60	Linksys 942	ed, D/Paulo	<input checked="" type="checkbox"/>

Abaixo do formulário, há um botão 'Inserir' e três links: 'Repor', 'Cancelar' e 'Ajuda'.

Figura 5.10 - Adicionar um terminal IP à plataforma

5.4.2 Função “Adicionar em massa”

Esta funcionalidade foi disponibilizada para permitir adicionar rapidamente vários dispositivos. Assim, nesta opção a plataforma solicitará apenas alguns parâmetros, tais como: o modelo do terminal IP, quantos terminais IP, e a primeira extensão a atribuir, como se pode observar na figura 5.11. De seguida, e depois de clicar no botão “Adicionar” é pedido por cada terminal o endereço MAC. Para facilitar esta tarefa, e no caso de serem muitos dispositivos a adicionar, pode-se utilizar um digitalizador de códigos de barras para ler o endereço MAC, que usualmente se encontra na própria caixa do equipamento ou nos manuais de utilização. Neste caso, é preciso ter o cuidado de configurar, no digitalizador, o carácter de mudança de linha (*enter*), após este ler o código, caso contrário este terá de ser pressionado manualmente.

A interface 'Adicionar em Massa' contém os seguintes campos:

Modelo do telefone: Polycom 330 Nº de telefones a adicionar: 25 Primeira extensão: 1001

Abaixo dos campos, encontra-se o botão 'Adicionar'.

Figura 5.11 - Adicionar terminais em massa

5.4.3 Função “Adicionados”

A função “Adicionados” efectua uma consulta à base de dados da plataforma de forma a devolver todos os terminais que foram adicionados. Na figura 5.12 pode-se visualizar esta função, onde constam várias informações sobre um terminal, tais como: extensão; senha; endereço MAC; modelo; localização; e se está ou não provisionado pela plataforma e a configuração no ficheiro `/etc/dhcpd.conf`.

Terminais adicionados à plataforma						
Ext	Senha	MAC	Modelo	Localização	Prov	Configuração DHCP
709997	709997	00:04:f2:1e:ac:44	Polycom 330	CI.Paulo Gomes	s	host Poly709997 { hardware ethernet 00:04:f2:1e:ac:44; }
709000	709000	00:04:f2:1e:a4:aa	Polycom 330	CI.Paulo Gomes	s	host Poly709000 { hardware ethernet 00:04:f2:1e:a4:aa; }
709996	709996	00:04:13:29:54:24	Snom 360	C2	s	host SNOM { hardware ethernet 00:04:13:29:54:24; }
709001	709001	00:04:f2:1e:ac:44	Polycom 330	CI-PGomes	s	host Poly709001 { hardware ethernet 00:04:f2:1e:ac:44; }
23989	23989	00:04:f2:1e:ac:44	Polycom 330	C2/PGomes	s	host Polycom { hardware ethernet 00:04:f2:1e:ac:44; }

Figura 5.12 - Terminais adicionados à plataforma

5.4.4 Função “Registados”

O objectivo desta função é devolver ao administrador da plataforma os terminais actualmente registados no IPPBX, ou seja, no Asterisk. Esta funcionalidade depende de uma rotina que é executada no IPPBX, criando uma lista dos terminais ligados que será disponibilizada via HTTP, ou copiada para o servidor de provisionamento via SCP. A figura 5.13 representa um exemplo da listagem de terminais registados, que pode ser filtrada por terminais ligados ou desligados.

Terminal IP Registados				
Filtro: Ligados Actualizar todos... Ligados Desligados				
	Extensao	Anfitriao	Porto	Estado
Terminal IP	102	192.168.69.15	5060	Ligado
Terminal IP	101	Desconhecido	0	Desligado
Terminal IP	100	Desconhecido	0	Desligado

Figura 5.13 - Terminais telefónicos IP registados

5.4.5 Função “Softphones”

Esta área da plataforma tem a finalidade de disponibilizar ao utilizador comum a possibilidade de descarregar um telefone virtual previamente provisionado, isto é, já com todas as configurações mais comuns efectuadas, com excepção da extensão/utilizador e palavra-passe. Uma vez que o X-lite apenas é livre para utilizadores domésticos, disponibilizou-se também o *softphone* Ekiga também provisionado, visto a licença deste ser de livre utilização e qualquer utilizador o pode usar. Na figura 5.14 apresenta-se o aspecto gráfico desta função.

Softphones	
X-Lite	
Descarregar	o telefone virtual (softphone) X-Lite.
Descarregar	o telefone virtual (softphone) X-Lite versão já provisionada.
Ekiga (provisionado)	
Descarregar	o telefone virtual (softphone) Ekiga.
Descarregar	o telefone virtual (softphone) Ekiga versão já provisionada.

Figura 5.14 - Softphones

5.4.6 Função “Agenda”

Esta funcionalidade da interface *web* tem o objectivo de activar a sincronização da agenda de contactos com os utilizadores da AD. Ao aceder à função “Agenda” surge

o ecrã apresentado na figura 5.15, onde é possível activar a sincronização com frequência diária, semanal e mensal. Ao serem activados um destes agendamentos a plataforma coloca um *script* no CRON que efectua via LDAP [Sermersheim06] consultas dos contactos à AD, converte no formato de agenda do telefone e o disponibiliza no directório do TFTP. No caso do Polycom, o ficheiro com os contactos é colocado na pasta pessoal do Polycom, uma vez que este utiliza o FTP para descarregar as configurações.

Ainda na figura 5.15, verifica-se que ao aceder a esta função é imediatamente apresentada a consulta feita à AD, onde são visíveis os nomes dos utilizadores, extensão telefónica, correio electrónico e localização.

Sincronização da agenda de contactos

☐ Diariamente
☐ Semanalmente
☐ Mensalmente
☐

Submeter

Nome	Apelido	Telefone	Correio Electrónico	Campus
Maria	Costa	3031	maria.costa@ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Marta	Henriques	3004	marta.henriques@ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Marta	Coutinho	3017	marta.coutinho@ipleiria.pt	SC - Serviços Centrais
Maurício	Sousa	3108	mauricio.sousa@ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Miguel	Silva	3022	miguel.silva@ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Miguel	Sampaio	3607	miguel.sampaio@ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Miguel	Sampaio	3607	miguel.n.sampaio@estg.ipleiria.pt	ESTG - Escola Superior de Tecnologia e Gestão
Mónica	Caldeira	3418	monica.caldeira@ipleiria.pt	SC - Serviços Centrais
Mónica	Felício	3037	monica.felicio@ipleiria.pt	SC - Serviços Centrais

Figura 5.15 - Sincronização da agenda de contactos

5.5 Grupo “Extensões”

Neste grupo de funcionalidades é possível efectuar algumas operações relacionadas com uma extensão. Visualizando a figura 5.2 verificam-se as quatro opções disponibilizadas nesta área.

5.5.1 Função “Detalhes”

A função “Detalhes”, além de mostrar todas as características de uma extensão, tais como: endereço IP, senha, endereço MAC, modelo de telefone associado, a sua localização, se está provisionada pela plataforma ou se está actualmente registada no Asterisk, permite também eliminar uma entrada ou alterar os seus dados.

Na figura 5.16 encontra-se um exemplo do detalhe de uma extensão adicionada na plataforma.

Extensão	Senha	MAC	Modelo	Localização	Provisionar
709001	12345	00:04:f2:1e:ac:44	Linksys 942	Campus2	<input type="checkbox"/>

[Repor](#) | [Cancelar](#) | [Ajuda](#)

Figura 5.16 - Detalhes de uma extensão

Esta função pode ser importante para o caso de ser necessário alterar o modelo de telefone associado a uma extensão. Neste caso, para além de se escolher o modelo do novo telefone seria necessário introduzir um novo endereço MAC, pois só assim é que o telefone iria ser reconhecido na plataforma de provisionamento.

5.5.2 Função “Reiniciar”

Esta área poderá ser bastante útil ao administrador da rede VoIP, pois poderá ser necessário reiniciar todos os terminais VoIP para actualizarem alguma configuração alterada na plataforma, ou os obrigar a actualizarem para um nova versão de *firmware*.

Na figura 5.17, estão disponíveis todas as opções desta função, onde se pode constatar que é possível reiniciar os terminais por modelo, por extensão, ou por endereço IP.



Figura 5.17 - Reiniciar terminais IP

5.5.3 Função “Monitor”

Esta funcionalidade serve para monitorizar as extensões provisionadas por esta plataforma, proporcionando uma forma visual onde seja fácil detectar uma extensão ou telefone sem conectividade. A figura 5.18 apresenta três extensões monitorizadas onde se verifica que a “709997” está desligada e as restantes estão conectadas à rede.



Figura 5.18 - Monitor de extensões

5.6 Grupo “Configurações em Massa”

Este grupo representa o maior objectivo desta plataforma de provisionamento, ou seja, permitir a possibilidade de configuração em massa de diversos parâmetros dos telefones IP. Analisando a figura 5.2 verifica-se que as opções disponíveis correspondem a cada dispositivo gerido pela plataforma, onde em cada uma é possível aceder aos parâmetros que podem ser personalizados em cada modelo, isto porque nem todos, por exemplo, suportam imagens de logotipo.

5.6.1 Função “Polycom 330”

Na área “Configurações em Massa”, estão disponíveis as funções que permitem personalizar cada telefone IP, em que apenas o Grandstream BT200 não possui todas as opções disponíveis, nomeadamente, “Imagem” e “Agenda”.

Passando a analisar a figura 5.19, verifica-se a capacidade de seleccionar outra versão *firmware*, outro toque de chamada e outra imagem de logotipo. É também possível, escolher o idioma por omissão e indicar que a agenda é centralizada, e com isto disponibilizar um ficheiro com os contactos importados da *Active Directory* e o enviar para os telefones, quando estes forem provisionados. Por último, possibilita uma funcionalidade importante que consiste em eliminar a senha de administração, definida por omissão na fábrica, para uma à escolha do administrador.



Figura 5.19 - Configurações em massa

Quando se editou as configurações, a selecção dos parâmetros é efectuada à custa de menus do tipo *drop down list*. Os ficheiros de toque disponíveis são provenientes do directório do Polycom, onde este vai listar todos os ficheiros com extensão *wav* sendo o processo semelhante para as imagens de logotipo e idioma. Já a opção “Agenda” disponibiliza a escolha entre agenda “Centralizada” e agenda “Local”.

5.7 Plano de Testes

Nesta secção é apresentado um plano de testes efectuado na plataforma de provisionamento. Serão apresentados testes de carga aos serviços de rede mais importantes, tais como: DHCP, TFTP, FTP e HTTP. Serão também expostos testes de funcionamento tendo em conta várias situações possíveis de ocorrer.

Na tabela 5.2 é apresentado resumidamente o plano de testes realizado aos componentes de rede, do servidor da plataforma de provisionamento. Foram feitos testes de carga, com ferramentas de injeção de tráfego, nos portos de rede de cada serviço. No caso de serviço DHCP foi ainda importante medir os pedidos de

configuração de rede (IP, máscara, porta de ligação, etc) e verificar as respectivas respostas.

Componente	Tipo de teste	Análise
DHCP	Testes de carga	Relatórios do sistema operativo de forma a verificar a resposta do servidor a todos os pedidos de configurações de rede. Desempenho da placa de rede, processador e memória do servidor.
TFTP	Testes de carga	Desempenho da placa de rede, processador e memória do servidor.
FTP	Testes de carga	Desempenho da placa de rede, processador e memória do servidor.
HTTP	Testes de carga	Desempenho da placa de rede, processador e memória do servidor.
Provisionamento	Provisionamento de terminais IP durante os testes de carga	Verificação do processo de provisionamento nos terminais.

Tabela 5.2 - Plano de testes aos vários componentes do servidor de provisionamento

Para os testes sobre o serviço DHCP utilizou-se a ferramenta Yersinia¹ disponível em variadíssimas distribuições de Linux. Esta ferramenta gera datagramas com origem em variados endereços MAC de forma a solicitar pedidos de configurações de rede (DHCP Discover) ao serviço DHCP da rede onde se encontra.

Já os testes de carga aos componentes TFTP, FTP e HTTP foram feitos com um *script* construído em Python² por Eli Fulkerson³ – `udp_stress_client.py`. Este script envia pacotes UDP para um determinado IP direccionado a um determinado porto. O tamanho e número dos pacotes é parametrizável quando se executa o *script*. Para alterar o IP e o porto é necessário editar o *script*.

As medições de desempenho foram feitas na infra-estrutura de virtualização dos Serviços Informáticos do IPL, onde está alojado o servidor de provisionamento.

Quanto a testes de funcionamento, estes estão resumidos na tabela 5.3. Estes testes reciam essencialmente situações que normalmente acontecem no dia-a-dia do

¹<http://sourceforge.net/projects/yersinia>

²<http://www.python.org>

³<http://www.elifulkerson.com>

utilizador destes sistemas de comunicação. Irão testar-se falhas de conectividade do servidor de provisionamento e do comutador de rede, quebras de serviços de rede e ocorrências nas ligações dos telefones IP à rede de dados e eléctrica.

Teste	Análise
Desligar um telefone IP e voltar a ligar.	Verificar o que acontece caso alguém desligue o telefone IP da tomada eléctrica.
Desligar um telefone IP da tomada de rede e voltar a ligar.	Verificar o que acontece caso alguém desligue o telefone IP da tomada de rede.
Mudar um telefone de tomada de rede.	Verificar o que acontece caso alguém mude o telefone IP de tomada de rede.
Falha de conectividade no <i>switch</i> .	Verificar o que acontece aos telefones IP quando se dá uma falha de conectividade com o comutador de rede (<i>switch</i>).
Reinício do servidor de provisionamento.	Verificar o que acontece aos terminais IP em funcionamento quando surge uma falha por alguns instantes com o servidor.
Serviço DHCP parado.	Verificar o comportamento dos telefones que estão ligados e a telefones que vão arrancar sem o serviço de DHCP.
Serviço TFTP, FTP e HTTP parado.	Verificar o comportamento dos telefones ao arrancarem sem as componentes que possibilitam o provisionamento.
Falha prolongada do servidor de provisionamento.	Verificar o que acontece com os telefones IP que já se encontravam operacionais. Verificar o que acontece com os telefones IP que foram ligados durante a falha.

Tabela 5.3 - Testes de funcionamento

5.7.1 Testes de carga

O teste de carga efectuado ao DHCP consistiu no envio de quantidades enormes de pedidos de configuração de rede (DHCP *Discover*) e aguardar a resposta que consistiu na oferta de endereço IP (DHCP *Offer*) e restantes configurações.

De acordo com a figura 5.3 num intervalo de apenas dois segundos foram pedidos 238 (barras azuis) IP e atribuídos os mesmos 238. As barras vermelhas significam informações que já não existem endereços livres para atribuir (*No free leases*).

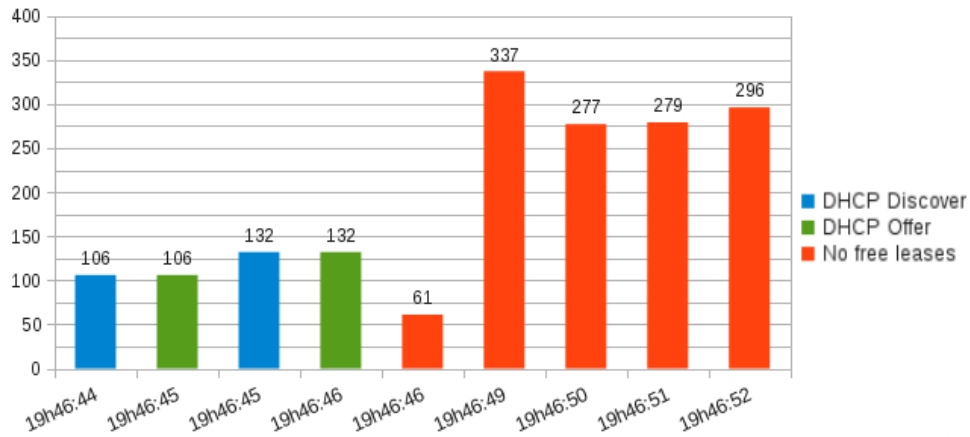


Figura 5.20 - Teste de carga ao DHCP

Enquanto decorriam os testes de carga ao DHCP, registou-se o desempenho da interface de rede, do processador e da memória.

Na figura 5.21 pode-se verificar que o tráfego da interface teve um pico máximo de cerca de 600 Kbps, muito abaixo dos 100Mbps que é a ligação à infra-estrutura de rede.

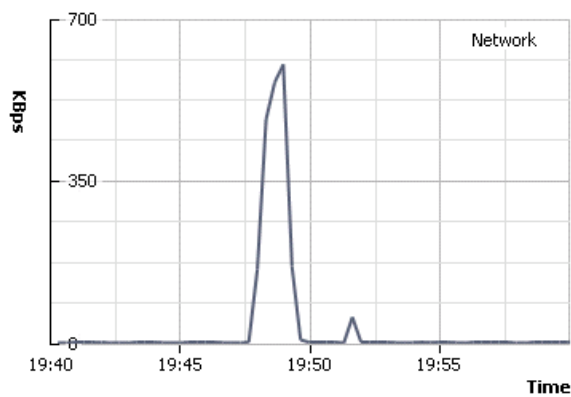


Figura 5.21 - Desempenho da interface de rede

Sendo 2.8GHz a velocidade máxima do processador pode-se verificar, depois de analisar a figura 5.22, que o pico de quase 600 MHz é muito abaixo do que o suportado.



Figura 5.22 - Desempenho do processador

Em termos de desempenho de memória, este é mostrado na figura 5.23 onde se verifica que o pico de memória utilizado rondou os 220 Mbytes (225000 Kbytes), metade da memória total do servidor de provisionamento.

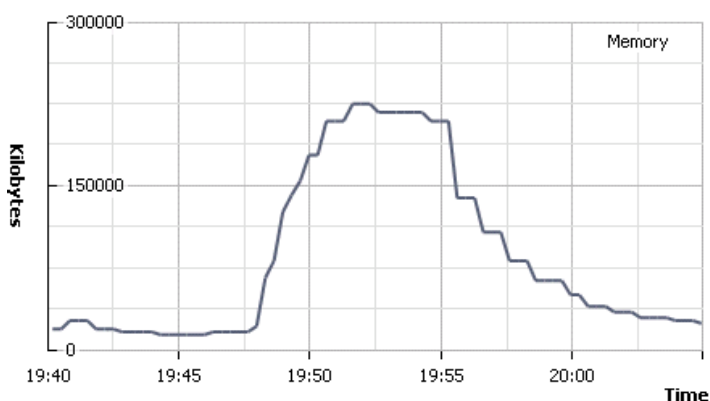


Figura 5.23 - Desempenho da memória

Conforme especificado no plano de testes, os serviços TFTP, FTP e HTTP foram submetidos aos testes de carga pelo *script* `udp_stress_client.py`. Na figura 5.24 apresenta-se a utilização do *script* onde realça o IP a testar, o porto de rede 21 (FTP), o tamanho (1000) dos pacotes e a quantidade (10000000) de pacotes a enviar. Os asteriscos correspondem a cada pacote enviado. Este *script* foi corrido em dois computadores com o objectivo de gerar mais tráfego para o servidor de

provisionamento. Foram injectados ao todo 60000 pacotes de 1 Mbyte em simultâneo nos portos 69 (TFTP), 21 (FTP) e 80 (HTTP).

```

root@PC01:/home/root/python udp_stress_client.py
-----
udp_stress_client.py
Updates and documentation (if any) at http://www.elifulkerson.com
-----
Starting client end. Control-C to quit.
Our target:
udp_stress_server.py running on 192.168.0.100 port 21
Enter number of bytes to send and the number of times to send them:
(for instance '100 10' to send 10 bursts of 100 bytes each)
% 1000 10000000
*****
*****

```

Figura 5.24 - Script `udp_stress_client.py`

Como resultado destes testes de carga apresentam-se os gráficos ilustrados nas figuras seguintes.

Na figura 5.26 verifica-se que o tráfego na interface de rede aumentou significativamente durante o envio de pacotes, contudo não foi suficiente sobrecarregar o servidor.

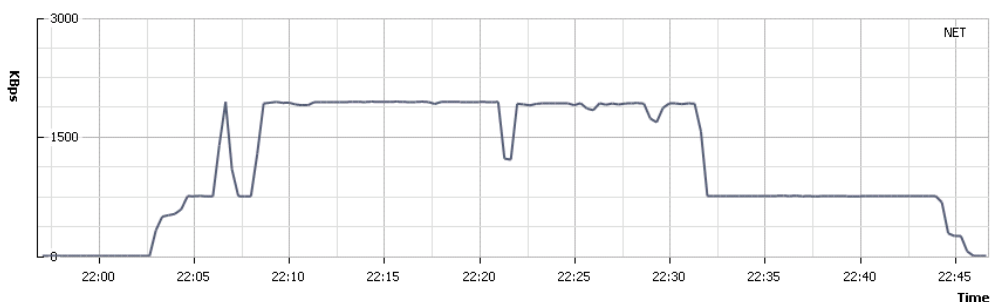


Figura 5.25 - Desempenho da interface de rede

O gráfico da figura 5.26 mostra a ocupação de processador durante os testes de carga e pelo que se pode apurar, os 60000 pacotes não foram suficientes para levar o servidor ao limite em termos de processamento.



Figura 5.26 - Desempenho do processador

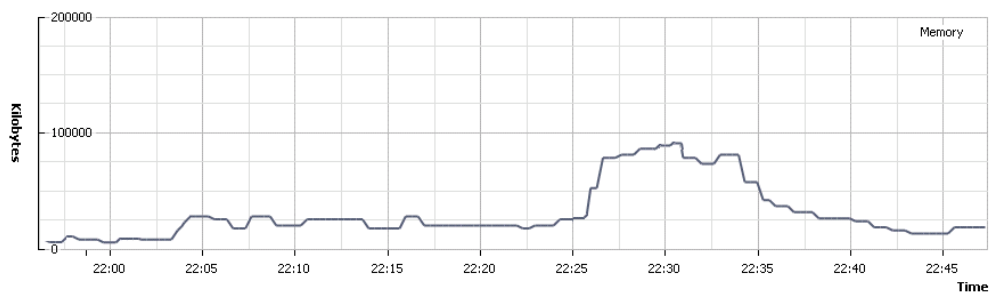


Figura 5.27 - Desempenho de memória

No gráfico de desempenho da memória verifica-se, à semelhança dos anteriores, que a ocupação fica abaixo ao limite do servidor. No entanto nota-se uma grande subida a partir das 22h55, que correspondem ao momento de arranque de quatro telefones IP ao mesmo tempo que decorria os testes de carga. Constata-se que com o aumento de de ocupação de memória verificado, poderá indiciar a necessidade de atribuição de pelo menos mais 512 Mbytes ao servidor. Como o servidor de provisionamento está assente numa plataforma de virtualização, a qualquer momento os SI do IPL podem aumentar este número.

5.7.2 Testes de Funcionamento

Nesta secção será apresentada a análise efectuada aos testes de funcionamento descritos na tabela 5.3.

i) Desligar um telefone IP e voltar a ligar

O primeiro teste realizado foi tão simples como desligar um telefone IP e voltar a ligar. Verificou-se que o arranque do terminal decorria normalmente e depois de alguns segundos ficava operacional e pronto a ser utilizado.

ii) Desligar um telefone IP da tomada de rede e voltar a ligar

Verificou-se que todos os telefones IP indicavam no visor que houve uma falha de ligação à rede. Ao voltar a ligar o cabo, verificou-se que bastavam poucos segundos, para os terminais voltarem a obter IP e ficarem operacionais.

iii) Falha de conectividade no switch

Experimentou-se desligar o *switch* e voltar a ligar. Verificou-se que o resultado foi semelhante ao ponto anterior, mas desta vez afectando todos os telefones IP.

iv) Reinício do servidor de provisionamento

Os resultados de um reinício do servidor de provisionamento são semelhantes aos dois testes anteriores. Neste o sistema demorou um pouco mais a voltar à normalidade devido ao tempo que o servidor demora a arrancar, que é aproximadamente um minuto.

v) Serviço DHCP parado

Verificou-se que ao parar o serviço de DHCP, os terminais que já estavam em funcionamento não foram afectados. Ao iniciar os quatro modelos de telefones IP testados, comprovou-se que nenhum iniciava devido a não conseguirem obter endereço IP.

vi) Serviço TFTP, FTP e HTTP parado

Com os serviços TFTP, FTP e HTTP parados os telefones IP, já em funcionamento, não sofreram qualquer problema. Testou-se o reinício dos quatro modelos e verificaram-se ocorrências distintas entre eles.

O Linksys SPA942 arrancou sem qualquer problema, com a configuração local ficando apto a ser utilizado.

O Polycom 330 devolveu a mensagem de erro “*Could not contact boot server. Booting exist configuration*” e depois de alguns segundos acabou por iniciar e ficar operacional com a configuração local.

O Snom 360 depois de cerca de 60 segundos deu a seguinte mensagem “*Repeating Prov.Request:5*”. Depois de repetir cinco vezes (*Prov.Request*) mostrou no visor outra mensagem “*Provision server failed*” e passados bastantes segundos acabou por arrancar com a configuração correcta, ficando operacional para executar a sua função.

O Grandstream BT200 arrancou sem qualquer problema e ficou apto a ser utilizado.

Resumindo, todos os telefones tinham sido provisionados, logo já tinham as configurações guardadas localmente, e apesar do servidor ter os serviços necessários ao provisionamento parados, estes arrancaram com as ultimas configurações provisionadas. De notar que dois dos modelos precisaram de mais tempo que o normal para o arranque do sistema.

vii) Falha prolongada do servidor de provisionamento

Efectuou-se um último teste que foi deixar o servidor de provisionamento desligado cerca de 24h. Verificou-se que todos os terminais que já estavam em funcionamento não tiveram qualquer problema. Em todos aqueles que arrancaram com o servidor de provisionamento desligado, aconteceu o mesmo que no teste do serviço DHCP parado, ou seja, não arrancaram por não conseguirem obter endereço IP.

5.8 Síntese

Este protótipo é o resultado prático desta dissertação, onde se começou pela definição e construção do cenário a utilizar, onde se instalaram e configuraram os vários componentes: Sistema Operativo Linux, serviço DHCP, TFTP, FTP, HTTP, e PHP.

Depois de todos os serviços essenciais em funcionamento, efectuaram-se testes de provisionamento, ainda num patamar a baixo nível, ou seja, na *bash* do sistema operativo, tendo em conta o estudo apresentado no capítulo 3. Posteriormente, automatizaram-se os processos mais úteis e mais utilizados numa consola *web*.

É de salientar que esta plataforma é um protótipo que futuramente se pretende completar e otimizar. Algumas das funções implementadas poderão ser melhoradas, como por exemplo as listagens de terminais registados, em que seria proveitoso cruzar esta informação com a base de dados (terminais adicionados) e com a monitorização, de forma a ter numa única listagem toda a informação. Pelo mesmo motivo, não se desenvolveram mecanismos de autenticação capazes de diferenciar utilizadores a áreas de administração e a áreas de utilizador. Caso a plataforma venha a ser utilizada por utilizadores, é conveniente apostar em mecanismos de autenticação, provavelmente integrados com a *Active Directory* do IPL.

O plano de testes consolidou a avaliação da plataforma com testes de carga ao servidor na tentativa de antecipar eventuais problemas de resposta em termos de rede, processador e memória.

Por fim, realça-se a importância da plataforma no seu todo, isto é, a plataforma é tanto constituída pela interface *web* como pela consola do sistema operativo, onde se podem realizar numerosas alterações, directamente nos ficheiros de configuração dos telefones, conforme documentado no capítulo 3.

6 Conclusões

Os principais objectivos deste trabalho foram o de especificar e conceber um modelo de plataforma para gestão de telefones IP, permitindo assim efectuar diversas operações através de um único ponto central. Sendo que esta plataforma iria servir como ferramenta de suporte e manutenção pelos administradores a todo o parque VoIP do IPL.

Como ponto de partida para atingir estes objectivos, procedeu-se no capítulo 2 à recolha de informação sobre soluções existentes na área, bem como uma pesquisa conceptual às tecnologias inerentes e consolidação de alguns conceitos.

Para perceber realmente quais as funcionalidades que o modelo deveria contemplar, efectuou-se, no capítulo 3, um estudo do provisionamento de cada telefone IP. Neste capítulo, para além de se obter um conhecimento do funcionamento de todos os dispositivos, também serviu para verificar na base do provisionamento quais as funções que estes suportavam.

No capítulo 4, foi definido uma arquitectura de provisionamento genérica tendo em conta o estudo apresentado no capítulo 2 e 3. Inicialmente, definiram-se os requisitos mais importantes que a plataforma deveria de ter implementado para servir de base às funções que a arquitectura definiu. O modelo de arquitectura (secção 4.2) caracterizou em forma de camadas a interoperabilidade entre o dispositivo VoIP e a interface de administração *web*. Estas camadas foram também dissecadas nas secções que se seguiram com a finalidade de detalhar o que acontece em cada fase.

A vantagem e utilidade desta abordagem em termos de arquitectura consiste em auxiliar a construção de uma plataforma, num formato *web* ou outro, a qualquer pessoa que consulte este documento.

No capítulo 5 para além da construção de um protótipo funcional pretendeu-se também de certa forma comprovar em ambiente laboratorial a arquitectura proposta. Ainda no mesmo capítulo elaborou-se um plano de testes ao protótipo da plataforma proposta, de forma a detectar eventuais problemas de resposta da mesma, quando confrontada com excesso de carga. Estes testes puderam comprovar que os componentes de rede (DHCP, TFTP, FTP e HTTP) que compõe a plataforma, bem como o hardware que a suporta foram adequadamente dimensionados.

Por fim, espera-se com este trabalho demonstrar o que é possível fazer na área do provisionamento de terminais de comunicação em VoIP, e contribuir para a implementação de uma solução no Instituto Politécnico de Leiria.

6.1 Principais Contribuições

Tomando em consideração os objectivos iniciais e os resultados obtidos, são agora sumariadas as principais contribuições deste trabalho:

- Pesquisa e análise de plataformas de provisionamento já existentes;
- Abordagem às funcionalidades de provisionamento suportados por cada modelo de terminal IP estudados;
- Proposta de uma arquitectura de provisionamento;
- Elaboração de um protótipo baseado na arquitectura proposta;
- Avaliação dos resultados extraídos dos testes efectuados ao protótipo em ambiente laboratorial.

6.2 Tópicos para Trabalho Futuro

Ao nível do protótipo aqui apresentando poderão desenvolver-se novas funcionalidades tais como:

- Mecanismos de autenticação capazes de diferenciar utilizadores a áreas de administração e a áreas de utilizador;
- Acrescentar novos provisionamentos de telefones IP baseado na importação de utilizadores da AD;
- Provisionamento dos telefones VoIP sem fios recentemente adquiridos pelo IPL;
- Ao criar um novo provisionamento, haver a possibilidade da plataforma criar a extensão no IPPBX Asterisk;
- Incorporar na plataforma web uma *softphone web* na área utilizador, onde este possa efectuar chamadas em qualquer sítio recorrendo unicamente a um navegador de Internet;
- Criação de grupos de terminais para possibilitar aplicação de configurações, personalizações e actualizações baseados nestes grupos;
- Desenvolver um sistema de tolerância a falhas já previsto na arquitectura apresentada neste trabalho;

Outra proposta que indirectamente poderá dar continuidade a este trabalho, será desenvolver uma proposta de um padrão de provisionamento e apresenta-lo a um consórcio de fabricantes, de forma a proporcionar métodos comuns de provisionar terminais IP.

Com a escassez de endereços IPv4 torna-se, num futuro não muito distante, converter as infra-estruturas para IPv6. Esta plataforma foi desenvolvida sobre a rede IPv4, assim, seria interessante desenvolver um trabalho que estude o provisionamento de terminais VoIP sobre infra-estrutura IPv6.

Referências

- [Chong04] Hui Min Chong e H. Scott Matthews, “Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems”, 2004.
- [Pires06] João J. O. Pires, 2006, “Sistemas e Redes de Telecomunicações”.
- [Anderson06] L. Andersson, Ed., E. Rosen, Ed., “Framework for Layer 2 Virtual Private Networks (L2VPNs)”, RFC 4664, Setembro 2006.
- [Florin10] Dalia Petcu, Ioan David e Florin Suci, “Globalization, Internet and Digital Divide”.
- [Maximilian10] Maximilian Ronniger, Florian Fankhauser, Christian Schanes, Thomas Grechenig, “A Robust and Flexible Test Environment for VoIP Security Tests”, 2010.
- [Silva03] Arlindo Maia da Silva, “Perspectivas de evolução de VoIP na Internet”, 2003.
- [Fernandes02] Gabriel Paiva Fernandes, “Implementação em DSP de um sistema Real Time Analyzer”, 2002.
- [Tittel96] Ed Tittel e Dawn Rader , “Computer telephony – Automating home office and small business”, 1996.
- [Droms97] R. Droms, “Dynamic Host Configuration Protocol”, RFC 2131, Março 1997.
- [Bathrick99] G. Bathrick, F. Ly, “Definitions of Managed Objects for the ADSL Lines”, RFC 2662, Agosto 1999.
- [Silva03] Arlindo Maia da Silva, “Perspectiva de evolução de VoIP na Internet”, 2003.
- [Cardoso06] Paulo Cardoso, “Assistente pessoal na selecção e utilização de serviços VoIP”, 2006.
- [Solari97] Stephen J. Solari, “Video and Audio Compression”, 1997.
- [Sahabudin09] Shahrizal Sahabudin e Mohamad Yusoff Alias, “End-to End Delay Performance Analysis of Various Codecs on VoIP Quality of Service”, 2009.
- [Minoli98] Daniel Minoli e Emma Monoli, “Delivering Voice over IP Networks”, 1998.
- [Rodman08] Jeffrey Rodman, “VoIP to 20 kHz: Codec Choices for High Definition Voice Telephon”, 2008.
- [Macian08] Carlos Macián, “How to Tune VoIP Codec Selection in WLANs”, Agosto 2008.
- [Gonia10] Karie Gonia, “Latency and QoS for Voice over IP”, 2010.
- [Gibson05] Jerry D. Gibson, “Speech Coding Methods, Standards, and Application”, 2005.

- [ITU10] <http://www.itu.int>, sítio do Web International Telecommunication Union, consultado em Janeiro de 2010.
- [Handley99] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, “SIP: Session Initiation Protocol”, RFC 2543, Março 1999.
- [Sinnreich01] Henry Sinnreich e Alan B. Johnston, “Internet Communications using SIP”, 2001
- [Spencer10] M. Spencer, B. Capouch, E. Guy, Ed., F. Miller, K. Shumard, “IAX: Inter-Asterisk eXchange Version 2”, RFC 5456, Fevereiro 2010.
- [Johnston04] Alan B. Johnston, “SIP: understanding the Session Initiation Protocol”, 2004.
- [Handley99] Handley, M., et al., “SIP: Session Initiation Protocol”, RFC 2543, 1999.
- [Rosenberg02] RFC 3261 .
- [Abbassi06] Abbassi, T., “A comparative study of the SIP and IAX VoIP protocols”, 2006.
- [Chava08] [Chava08] – Chava, K.S.; How, J., “Integration of Open Source and Enterprise IP PBXs”, 2008.
- [Farrell09] Amy Farrel, “X-Lite Softphone reaches 340000 downloads per month”, 2009, consultado na B-ON.pt em Fevereiro de 2010.
- [Andre04] P. Saint-Andre, Ed., “Extensible Messaging and Presence Protocol (XMPP): Core”, Outubro 2004.
- [EK10] Ekiga, “Documentação do fabricante“, <http://wiki.ekiga.org/index.php/Documentation>, consultado em Fevereiro de 2010.
- [Robert08] Likic Robert, Bosnjak Darko, Maxwell Simon, Marinovic Darko, Francetic Igor e Cikes Nada, “To VoIP or not to VoIP, is this really the question now?”, 2008.
- [NE96] Microsoft, “Network Essentials – Hands-On, Self-Paced Training for Supporting Local and Wide Area Networks”, 1996.
- [Schwarz04] Brett Schwarz, “Asterisk Open-Source PBX System”, Fevereiro 2004.
- [CiscoAVVID10] http://www.cisco.com/application/pdf/en/us/guest/tech/tk621/c1501/ccmigration_09186a0080174993.pdf, sítio web consultado em Fevereiro de 2010.
- [Jason05] O'Grady, Jason D., “Dialing for less”, 2005.
- [Rodrigues06] Nuno G. Rodrigues e Albano A. G. Alves, “Implementação de Serviços de Telefonia IP numa Instituição de Ensino Superior”, 2006.
- [Wong06] William Wong, “Open-Source Platform Dials Into VoIP”, 2006.
- [Goel09] Saurabh Goel (IIITM - Indian Institute of Information Technology and Management), “ASR System Integration with Asterisk for SIP or IAX Softphone Clients”, 2009.
- [Yang09] Xiaohui Yan, “A society integrated tested architecture for peer-to-peer telecommunications”, 2009.
- [CiscoUPM09] Cisco Unified Provisioning Manager, “Supported Devices Table for Cisco Unified Provisioning Manager 1.3.1”, Janeiro 2009, http://www.cisco.com/en/US/docs/net_mgmt/cisco_unified_provisioning_manager/1.3.1/device_support/table/CUPMdvsp.html, Sítio web consultado em Abril de 2010.
- [Johnston06] M. Johnston e S. Venaas, Ed., “Dynamic Host Configuration Protocol (DHCP)

- Options for the Intel Preboot eXecution Environment (PXE)", RFC4578, Novembro 2006.
- [Postel85] J. Postel, J. Reynolds, "File Transfer Protocol (FTP)", RFC 959, Outubro 1985
- [Fielding99] R. Fielding, UC Irvine, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach e T. Berners-Lee "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, Junho 1999.
- [Yang10] Yun Yang e Jia Mi, "Design of DHCP Protocol based on Access Control and SAKA Encryption Algorithm", 2010.
- [Connor04] Owen O'Connor, "Deploying forensic tools via PXE", 2004.
- [Baker04] M. Baker, M. Nottingham, "The application/soap+xml media type", RFC 3902, Setembro 2004.
- [Srinivasan95] R. Srinivasan, "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 1057, Agosto 1995.
- [BF07] Broadband Forum, "TR-069 CPE Wan Management Protocol v1.1 Amendment 2", 2007.
- [BDF10] "Broad Band Forum", <http://www.broadband-forum.org>, sítio web consultado em Fevereiro de 2010.
- [TB10] "Trixbox Pro Feature Descriptions", <http://www.trixbox.com/products/trixbox-pro/feature-descriptions/>, sítio web oficial da distribuição Trixbox, consultado em Fevereiro de 2010.
- [Sollins92] K. Sollins, "The TFTP Protocol (Revision 2)", RFC 1350, Julho 1992.
- [Ray03] Erik T. Ray, " Learning XML", 2003.
- [PL10] "White Paper, Configuration File Management on SoundPoint IP Phones", http://www.polycom.com/common/documents/whitepapers/configuration_file_management_on_soundpoint_ip_phones.pdf, sítio web consultado em Abril de 2010.
- [Josefsson06] S. Josefsson, "The Base16, Base32, and Base64 Data Encodings", RFC 4648, Outubro 2006.
- [Jacobs07] SPA942 Personal Directory, Geoff Jacob, 2007 - <http://blog.grimsy.net/2007/02/23/spa942-personal-directory>, consultado em Abril de 2010.
- [Paulie10] Paulie, "Simple GUI for Linksys SPA 962 / SPA942 / SPA941 Personal Directory", <http://www.tachytelic.net/2010/05/simple-gui-for-linksys-spa942-personal-directory/>, consultado em Junho de 2010.
- [GRT10] Grandstream, "Custom Ringtones", <http://www.grandstream.com/faqscustomringtones.html>, sítio web consultado em Abril de 2010.
- [Dierks99] T. Dierks, C. Allen, "The TLS Protocol Version 1.0", RFC 2246, Janeiro 1999.
- [Sermersheim06] J. Sermersheim, Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, Junho 2006.
- [Membrey09] Peter Membrey, Tim Verhoeven and Ralph Angenendt, "The complete guide to network administration using CentOS", 2009.
- [Turnbull09] James Turnbull, Peter Lieverdink and Dennis Matotek, "Pro Linux System Administration", 2009.
- [Brand05] Jason Brand e Jeff Balvanz, "Automation is a Breeze with Autoit", 2005.

Anexos

A - Ferramentas do telefone IP BT200

i) Granstream Configuration Generator

Esta ferramenta permite gerar ficheiros de configuração binários a partir do ficheiro de texto já previamente criado - **x500_bt200.cfg**.

Como mostra a figura 2, na caixa de texto “*listing*” coloca-se o endereço MAC do telefone em causa, no campo “*Configuration Template*” escolheu-se o ficheiro de configuração em modo de texto e, no campo “*Destination Path*” escolhe-se a localização do ficheiro de destino.

De seguida, pressiona-se o botão “*Go*” para se obter o ficheiro de configuração binário com o nome **cfg000b820cf476**. Por sua vez este ficheiro deverá ser colocado na pasta **/tftpboot** do IPPBX ou do servidor de provisionamento. Depois de reiniciar o telefone, este ficará automaticamente com as configurações baseadas no ficheiro **cfg000b820cf476**.

Os restantes campos podem ou não ser preenchidos, conforme as configurações feitas no ficheiro de configuração inicial - **x500_bt200.cfg**.

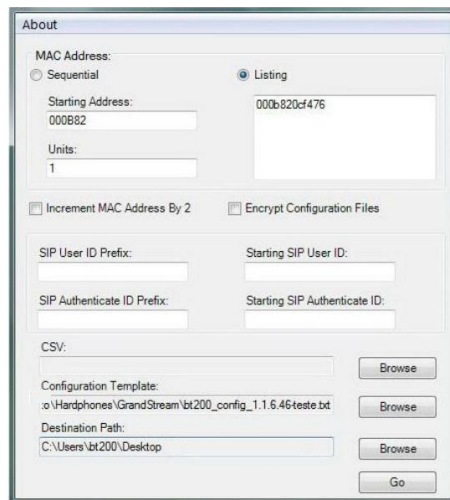


Figura 1 - Grandstream Configuration Generator

ii) Ringtool

A ferramenta *Ringtool*, disponibilizada pelo fabricante, serve para converter um ficheiro áudio para as características suportadas pelo Grandstream BT200, ou seja, 16 bits com uma frequência máxima de 8kHz.

Analisando a figura 2 verifica-se que na conversão é possível escolher a duração do toque, e o nome do ficheiro de saída (**ring1.bin**, **ring2.bin** ou **ring3.bin**). O ficheiro de entrada pode ser escolhido no botão “*Select*” e tem de ter a extensão wav. Depois da conversão o toque deverá ser colocado na directoria /tftpboot do IPPBX ou servidor de provisionamento.

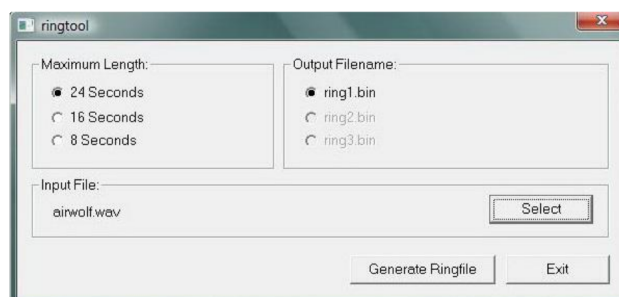


Figura 2 - Ringtool

B - Índices da Agenda do telefone SPA942

Nº de entrada	Índice	Nº de entrada	Índice
1	28398	2	28334
3	28526	4	28462
5	28654	6	28590
7	27758	8	27694
9	27886	10	27822
11	28014	12	27950
13	28142	14	28078
15	21102	16	21038
17	21230	18	21166
19	21358	20	21294
21	21486	22	21422
23	20590	24	20526
25	20718	26	20654
27	20846	28	20782
29	20974	30	20910
31	22126	32	22062
33	22254	34	22190
35	22382	36	22318
37	22510	38	22446
39	21614	40	21550
41	21742	42	21678
43	21870	44	21806
45	21998	46	21934
47	23150	48	23086
49	23278	50	23214
51	23406	52	23342
53	23534	54	23470
55	22638	56	22574
57	22766	58	22702
59	22894	60	22830
61	23022	62	22958
63	24174	64	24110
65	24302	66	24238
67	24430	68	24366
69	24558	70	24494
71	23662	72	23598
73	23790	74	23726
75	23918	76	23854
77	24046	78	23982
79	17006	80	16942
81	17134	82	17070
83	17262	84	17198
85	17390	86	17326
87	16494	88	16430
89	16622	90	16558
91	16750	92	16686
93	16878	94	16814
95	18030	96	17966
97	18158	98	18094
99	18286	100	18222

C - AutoIT - provisionamento do X-Lite

O AutoIT é uma ferramenta livre que automatiza tarefas num sistema operativo Windows. O seu modo de funcionamento é baseado num *script* que executa comandos. Uma das forma de construir estes *scripts* é através de cliques do rato combinando-os com instruções.

Depois da instalação da aplicação AutoIT Script, de seguida, executa-se o AutoIT *Window Info*, que deverá fazer surgir a janela apresentada na figura 3. Aí estará o “*Finder Tool*”(FT) que é uma funcionalidade essencial na automação de cliques do rato. Ao arrastar o FT para cima de uma janela, é mostrado no separador “*Control*” toda a informação sobre essa janela, tais como: o nome, identificador, classe, tamanho, entre outros.

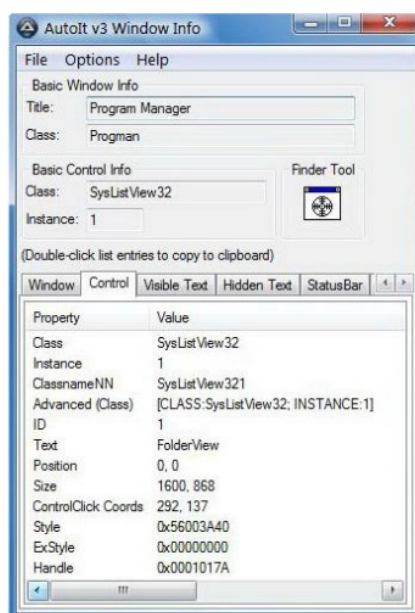


Figura 3 - AutoIT Windows Info

De seguida, deve-se correr a aplicação SciTE Script Editor e coloca-se o código apresentado na figura 4.

```
; Executar o instalador  
Run ( "X-Lite_Win32.exe" )
```

Figura 4 - Código para executar o instalador

Analisando a figura 4, a primeira linha de código serve para correr o instalador, fornecendo-se para tal o nome do mesmo.

Ao correr a aplicação será apresentada a primeira janela, a janela de boas vindas. Para seguir em frente, é necessário automatizar o clique do botão esquerdo do rato sobre o botão “Next”. Para tal, arrastando a mira da “Finder Tool” para cima do botão “Next”, produz-se no separador “Control” a informação sobre esse botão, como se por ver na figura 5.

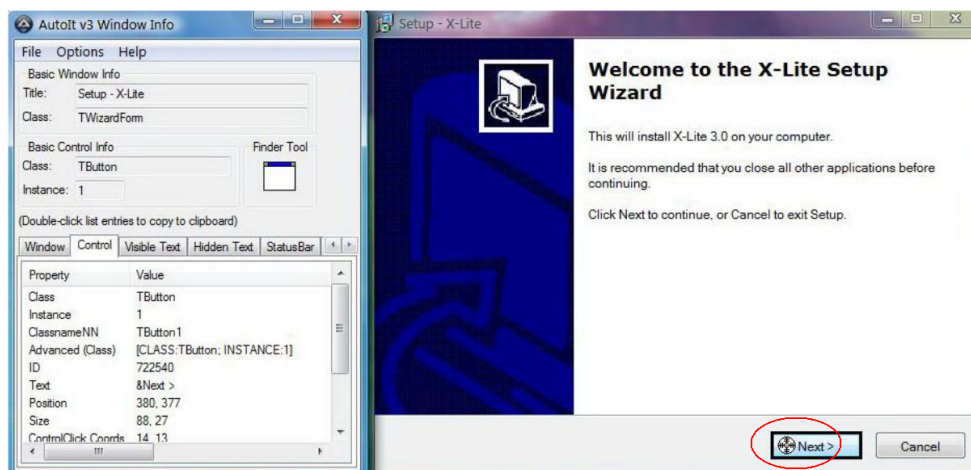


Figura 5 - Interação do "Finder Tool" com um Radio Button

No editor o código a adicionar é o apresentado na figura 6, onde a primeira linha vai tornar activa a janela com o nome “Setup – X-Lite”, a segunda faz uma pausa até esta janela surgir. A última linha é responsável por clicar no botão, e recebe como parâmetros o nome da janela, o nome do botão, o identificador do botão e o botão do rato que deve efectuar o clique, neste caso o esquerdo .

```

;Ecrã inicial
WinActivate ( "Setup - X-Lite" )
WinWait ( "Setup - X-Lite" )
ControlClick ( "Setup - X-Lite", "&Next >", "[CLASS:TButton; INSTANCE:2]", "left" )

```

Figura 6 - Ecrã inicial

A seguir surge a janela onde é necessário seleccionar “*I accept the agreement*”, como se pode ver na figura 7.

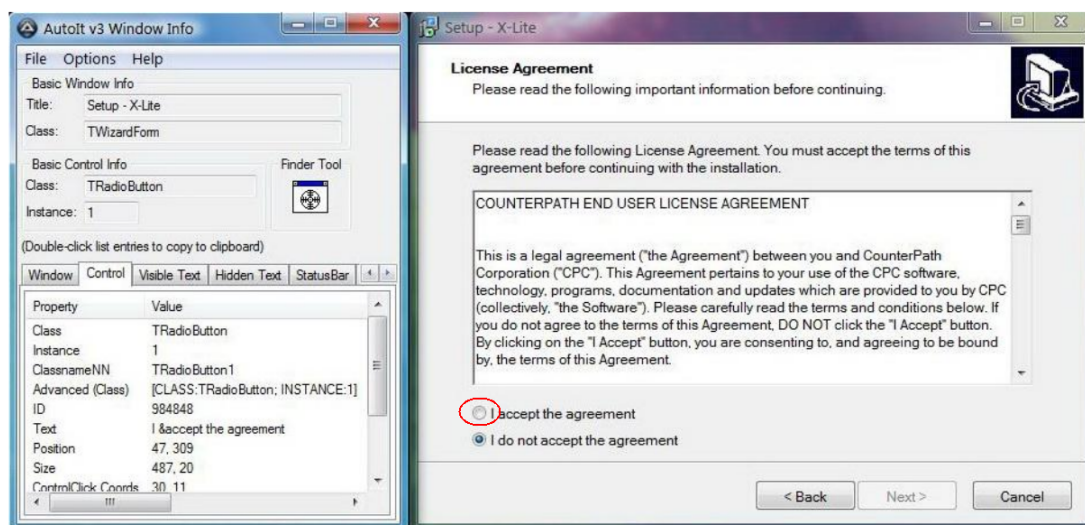


Figura 7 - Selecção do acordo de licença

Convertendo o processo de selecção do acordo de licença em código de AutoIT tem-se as instruções apresentadas na figura 8. Para que isto aconteça há que tornar a janela activa, ou esperar que apareça, usando as funções **WinActivate()** e **WinWait()**. De seguida, com a função **ControlCommand()** muda-se o estado do Radio Button para checked. Os argumentos recebidos por esta função são o nome da janela, o nome do campo, o identificador do campo obtido pela “Finder Tool” e se o que se pretende é fazer check ou uncheck ao botão. Finalmente, com a função **ControlClick()**, pressiona-se novamente o botão “Next”.


```

; Acordo de licença
WinActivate ( "Setup - X-Lite" )
WinWait ( "Setup - X-Lite" )
ControlCommand ( "Setup - X-Lite", "I &accept the agreement", "[CLASS:TRadioButton; INSTANCE:1]",
"Check", "" )
ControlClick ( "Setup - X-Lite", "&Next >", "[CLASS:TButton; INSTANCE:2]", "left" )

```

Figura 8 - Aceitação do acordo da licença

Para o resto da instalação, as opções serão todas por omissão, sendo assim, a instrução do AutoIT a utilizar será a **ControlClick()** para automatizar a passagem de janela em janela e até à conclusão da instalação, como se pode verificar na figura 9.

```

; Pasta de instalação
WinActivate ( "Setup - X-Lite" )
WinWaitActive ( "Setup - X-Lite", "Select Destination Location" )
ControlClick ( "Setup - X-Lite", "&Next >", "[CLASS:TButton; INSTANCE:3]", "left" )
; Selecção de outras opções
WinActivate ( "Setup - X-Lite" )
WinWaitActive ( "Setup - X-Lite", "Select Additional Tasks" )
ControlClick ( "Setup - X-Lite", "&Next >", "[CLASS:TButton; INSTANCE:3]", "left" )
; Conclusão da instalação
WinActivate ( "Setup - X-Lite" )
WinWaitActive ( "Setup - X-Lite", "Completing the X-Lite Setup Wizard" )
ControlClick ( "Setup - X-Lite", "&Finish", "[CLASS:TButton; INSTANCE:3]", "left" )

```

Figura 9 - Botão "Next" até ao final da instalação

Por fim, falta automatizar a configuração de registo do *softphone* X-Lite no IPPBX, permitindo que qualquer utilizador, apenas munido da sua extensão e senha, consiga colocar esta aplicação em funcionamento e comece desde já a efectuar chamadas.

Depois da instalação do X-Lite, este irá iniciar automaticamente, e surgirá a janela apresentada na figura 10, onde se configuram as contas SIP.

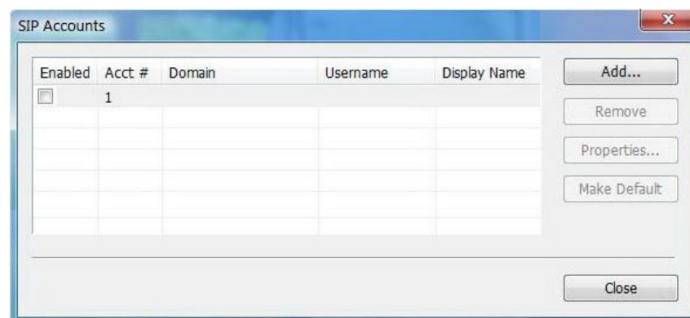


Figura 10 - Janela de Configuração da conta SIP no X-Lite

O processo seguinte será automatizar a criação de uma conta SIP no X-Lite. O código apresentado na figura 11 serve para parar a execução do *script* – **WinWait("SIP Accounts")**, até surgir a janela com o nome “SIP Accounts” para primeiro plano – **WinActivate ("SIP Accounts")**. A última instrução faz o clique no botão “Add” que faz com que surja a janela apresentada na figura 12.

```
; Conta SIP
WinWait( "SIP Accounts" )
WinActivate ( "SIP Accounts" )
ControlClick ( "SIP Accounts", "&Add...", "[CLASS:Button; INSTANCE:1]", "left" )
```

Figura 11 - Configuração da conta SIP

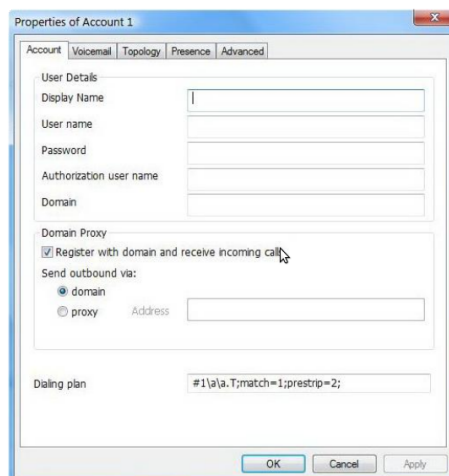


Figura 12 - Propriedades da conta SIP

Para que, durante a instalação, seja solicitado ao utilizador que introduza o seu nome de utilizador, a sua extensão telefónica e senha de autenticação, para efectuar o registo junto do IPPBX, foi gerado o seguinte código:

```
WinWaitActive ( "Properties of Account 1" )
ControlSetText ( "Properties of Account 1", "", "[CLASS:Edit; INSTANCE:5]", "192.168.0.100" )
$extensao = InputBox ( "Credenciais", "Insira o número da sua extensão telefonica:", "", "", -1, -1, "Default",
"Default" )
WinWaitClose ( "Credenciais", "Insira o número da sua extensão." )
$login = InputBox ( "Credenciais", "Insira o seu User Name:", "", "", -1, -1, "Default", "Default" )
WinWaitClose ( "Login", "Insira o seu User Name:" )
$password = InputBox ( "Credenciais", "Insira a sua Password:", "", "*", -1, -1, "Default", "Default" )
WinWaitClose ( "Password AD", "Insira a sua Password:" )
WinActivate ( "Properties of Account 1" )
WinWaitActive ( "Properties of Account 1" )
ControlSetText ( "Properties of Account 1", "", "[CLASS:Edit; INSTANCE:2]", $extensao )
ControlSetText ( "Properties of Account 1", "", "[CLASS:Edit; INSTANCE:3]", $password )
ControlSetText ( "Properties of Account 1", "", "[CLASS:Edit; INSTANCE:4]", $login )
sleep (3)
ControlClick ( "Properties of Account 1", "OK", "[CLASS:Button; INSTANCE:9]", "left" )
WinActivate ( "SIP Accounts" )
ControlClick ( "SIP Accounts", "Close", "[CLASS:Button; INSTANCE:5]", "left" )
```

Figura 13 - Introdução de dados de autenticação da conta SIP

A função **InputBox()**, irá fazer surgir três janelas a pedir dados. A primeira irá pedir a extensão telefónica que será guardada na variável **\$extensão**, a segunda irá pedir o nome de utilizador que será guardado na variável **\$login** e a terceira irá pedir ao utilizador a senha da sua conta SIP, que será guardada na variável **\$password**.

Com a função **ControlSetText()** e, novamente, com o “*Finder Tool*”, o valor de cada uma das variáveis será copiada para o campo respectivo na janela “*Properties of Account 1*” (figura 12).

Através da função **ControlSetText()**, o campo “*Domain*” será preenchido com o valor 192.168.0.100, correspondente ao endereço IP do IPPBX.

Por fim, resta simular os cliques sobre os botões “OK” e “Close” para fechar as janelas abertas.

Na figura 14, apresenta-se o código de uma janela a confirmar o sucesso da instalação através da primeira instrução em que o utilizador terá de clicar num botão “Ok” para fechar a mesma. Caso o utilizador não clique em “OK” a janela será

automaticamente fechada após 10 segundos (**WinWaitClose**). A última instrução faz o encerramento da aplicação.

```
MsgBox ( 0, "PARABENS", "X-Lite Instalado e Configurado com Sucesso.", 10 )  
WinWaitClose ( "PARABENS", "X-Lite Instalado e Configurado com Sucesso" )  
WinClose ( "X-Lite" )
```

Figura 14 - Ecrã final da instalação

Para finalizar toda esta operação, resta compilar o *script* gerado pelo AutoIT. Para este efeito, usa-se a ferramenta AutoIT Script to EXE Converter que é apresentada na figura 15.

Para compilar basta seleccionar o *script* gerado e o nome do ficheiro de saída e clicar no botão “Convert”.

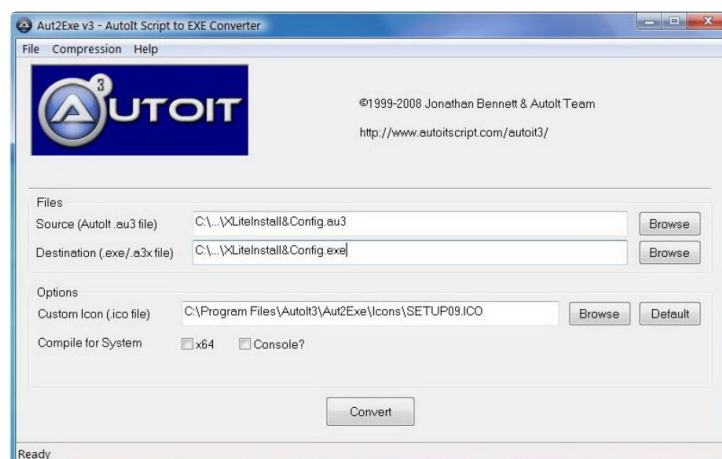


Figura 15 - Interface do AutoIT Script to EXE Converter

D - Ficheiro com idioma em Português do Snom 360

```
<?xml version="1.0" encoding="utf-8"?>
<phrases>
<phrase i="0" n="lang_none" t=""/>
<phrase i="1" n="lang_language_name" t="Portugues"/>
<phrase i="2" n="lang_cntry_CHN" t="China"/>
<phrase i="4" n="lang_cntry_GER" t="Alemanha"/>
<phrase i="10" n="lang_cntry_PRT" t="Portugal"/>
<phrase i="15" n="lang_cntry_CHL" t="Chile"/>
<phrase i="24" n="lang_cntry_ESP" t="Espanha"/>
<phrase i="25" n="lang_cntry_ALB" t="Albania"/>
<phrase i="28" n="lang_cntry_BRA" t="Brasil"/>
<phrase i="29" n="lang_cntry_CAI" t="Turks e Caicos Ilhas"/>
<phrase i="30" n="lang_cntry_CHA" t="Chatam"/>
<phrase i="95" n="lang_def_on" t="*Ligar"/>
<phrase i="96" n="lang_def_off" t="*Desligar"/>
<phrase i="97" n="lang_busy" t="Ocupado"/>
<phrase i="98" n="lang_def_busy" t="*Ocupado"/>
<phrase i="99" n="lang_deny" t="Negar"/>
<phrase i="100" n="lang_hold" t="Reter"/>
<phrase i="101" n="lang_transfer" t="Transferir"/>
<phrase i="102" n="lang_conference_on" t="Cnf.On"/>
<phrase i="103" n="lang_conference_off" t="Cnf.Off"/>
<phrase i="104" n="lang_cmc" t="CMC"/>
<phrase i="105" n="lang_edit_cmc" t="Client Matter Code:"/>
<phrase i="107" n="lang_del" t="Apagar"/>
<phrase i="109" n="lang_red" t="Redirecionar"/>
<phrase i="110" n="lang_missed" t="Perdido: "/>
<phrase i="111" n="lang_edit" t="Editar"/>
<phrase i="112" n="lang_mwi" t="Mensagem"/>
<phrase i="113" n="lang_dhcp" t="DHCP"/>
<phrase i="114" n="lang_enum" t="ENUM"/>
<phrase i="115" n="lang_deny_all" t="Block"/>
<phrase i="116" n="lang_headset_rj" t="HeadRJ"/>
<phrase i="117" n="lang_headset_pc" t="Head"/>
<phrase i="118" n="lang_mwi_mailbox1" t="Correio de:"/>
<phrase i="119" n="lang_mwi_mailbox2" t="Voce tem"/>
<phrase i="120" n="lang_mwi_mailbox3" t="Tem novas mensagens."/>
```

```

<phrase i="121" n="lang_mwi_new" t="novo"/>
<phrase i="122" n="lang_mwi_old" t="mensagens antigas"/>
<phrase i="123" n="lang_mwi_goto_mailbox" t="Ir para o correio?"/>
<phrase i="124" n="lang_challenge_user" t="User"/>
<phrase i="125" n="lang_challenge_pw" t="Password"/>
<phrase i="126" n="lang_vmail" t="Mensagem"/>
<phrase i="127" n="lang_sms" t="SMS"/>
<phrase i="128" n="lang_no_dialplan" t="Sem Dialplan"/>
<phrase i="129" n="lang_use_enum" t="Use ENUM?"/>
<phrase i="130" n="lang_area_code" t="Codigo de Area:"/>
<phrase i="131" n="lang_country_code" t="Codigo do Pais:"/>
<phrase i="132" n="lang_wiz_dhcp" t="Usar DHCP?"/>
<phrase i="133" n="lang_wiz_ip_adr" t="IP:"/>
<phrase i="134" n="lang_wiz_mac" t="MAC:"/>
<phrase i="144" n="lang_wiz_functionkey_number" t="Numero:"/>
<phrase i="145" n="lang_wiz_welcome" t="Benvindo! Pressione uma tecla para iniciar."/>
<phrase i="146" n="lang_terminated_finished" t="Finalizado: "/>
<phrase i="147" n="lang_try_parking" t="Fila de espera!"/>
<phrase i="156" n="lang_list_details_time" t="Tempo:"/>
<phrase i="157" n="lang_list_details_duration" t="Duracao:"/>
<phrase i="158" n="lang_list_details_costs" t="Custos:"/>
<phrase i="159" n="lang_list_details_missed" t="Perdida(s):"/>
<phrase i="160" n="lang_list_details_to" t="Para:"/>
<phrase i="161" n="lang_list_details_from" t="De:"/>
<phrase i="168" n="lang_reg_explanation_failed" t="Registo falhou"/>
<phrase i="169" n="lang_reg_explanation_closed" t="Registo fechado"/>
<phrase i="170" n="lang_reg_explanation_registering" t="Registando"/>
<phrase i="171" n="lang_reg_explanation_registered" t="Registado"/>
<phrase i="172" n="lang_reg_explanation_trying" t="Tentando registrar"/>
<phrase i="173" n="lang_dnd_line" t="N&#195;&#163;o Perturbar?"/>
<phrase i="174" n="lang_menu_redirect1" t="&#27; Evento Redirecionado &#26;"/>
<phrase i="198" n="lang_menu_vol_head_txt" t="Auricular"/>
<phrase i="199" n="lang_menu_vol_head" t="Auricular"/>
<phrase i="200" n="lang_menu_vol_mic" t="Microphone"/>
<phrase i="201" n="lang_menu_vol_speak" t="Viva-Voz"/>
<phrase i="202" n="lang_menu_gen_contrast" t="Contraste"/>
<phrase i="203" n="lang_menu_gen_lang" t="Idioma"/>
<phrase i="204" n="lang_menu_gen_ringtone" t="Toque"/>
<phrase i="205" n="lang_menu_web_webserver_type" t="Servidor"/>
<phrase i="206" n="lang_menu_web_http_port" t="HTTP"/>

```

```

<phrase i="207" n="lang_menu_web_https_port" t="HTTPS"/>
<phrase i="208" n="lang_menu_function_key" t="Fkeys"/>
<phrase i="209" n="lang_menu_headset_rj" t="On"/>
<phrase i="210" n="lang_menu_def_headset_rj" t="*On"/>
<phrase i="211" n="lang_menu_headset_none" t="Off"/>
<phrase i="212" n="lang_menu_def_headset_none" t="*Off"/>
<phrase i="213" n="lang_menu_headset_chinch" t="Chinch"/>
<phrase i="214" n="lang_menu_def_headset_chinch" t="*Chinch"/>
<phrase i="215" n="lang_menu_oth_zone" t="TimeZone"/>
<phrase i="216" n="lang_menu_oth_dialtone" t="Tons"/>
<phrase i="217" n="lang_menu_tser_alarm" t="Alarme"/>
<phrase i="218" n="lang_menu_tser_stopwatch" t="StopWatch"/>
<phrase i="219" n="lang_menu_tser_timer" t="Timer"/>
<phrase i="220" n="lang_menu_tser_alarm_time" t="Alarm time:"/>
<phrase i="224" n="lang_menu_red_time2" t="Redirecionar em:"/>
<phrase i="225" n="lang_menu_red_number" t="Always target:"/>
<phrase i="226" n="lang_menu_red_busy_number" t="Ocup. redirecionar para:"/>
<phrase i="227" n="lang_menu_red_target" t="Redirecionamento"/>
<phrase i="228" n="lang_menu_ring_normal" t="Normal"/>
<phrase i="229" n="lang_menu_ring_vip" t="VIP"/>
<phrase i="231" n="lang_menu_ring_select_vip" t="Seleccionar melodia VIP:"/>
<phrase i="232" n="lang_menu_ringer1" t="Toque 1"/>
<phrase i="233" n="lang_menu_ringer2" t="Toque 2"/>
<phrase i="242" n="lang_menu_vlan_id" t="ID"/>
<phrase i="243" n="lang_menu_vlan_qos" t="Prioridade"/>
<phrase i="244" n="lang_menu_vlan_reset" t="Reset"/>
<phrase i="245" n="lang_menu_system_net_info" t="Rede"/>
<phrase i="246" n="lang_menu_system_mem_info" t="Memoria"/>
<phrase i="247" n="lang_vol_headset" t="Volume do Auricular:"/>
<phrase i="248" n="lang_vol_handset" t="Volume do Telefone:"/>
<phrase i="249" n="lang_vol_speaker" t="Volume do Viva-Voz:"/>
<phrase i="250" n="lang_vol_contrast" t="Contraste do visor:"/>
<phrase i="251" n="lang_vol_mic" t="Volume do Microfone:"/>
<phrase i="258" n="lang_ringing" t="A chamar"/>
</phrases>

```

E - Opções de configuração do Grandstream BT200

```
#####  
## Configuration template for BT200 and BT201 firmware version 1.1.6.46  
#####  
  
#####  
## Advanced/System-wide Options  
#####  
# Enable Provider Lock. 0 - No, 1 - Yes.  
P9999 = 0  
# Provider Lock Key. A string of up to 16 bytes.  
P9998 =  
# Provider Authentication. A string of up to 16 bytes  
P9997 =  
# Password for configuration file authentication  
P1 =  
# Admin password for web interface  
P2 = {$admin_pass}  
# G723 rate  
# 0 = 6.3 kbps, 1 = 5.3 kbps  
P49 = 0  
# iLBC Frame Size. 0 - 20ms(default), 1 - 30ms.  
P97 = 0  
# iLBC payload type. Between 96 and 127, default is 97.  
P96 = 97  
# Silence Suppression. 0 - no, 1 - yes  
P50 = 0  
# Voice Frames per TX (up to 10/20/32/64 frames for G711/G726/G723/other codecs respectively)  
P37 = 2  
# Layer 3 QoS (IP Diff-Serv or Precedence value for RTP)  
P38 = 48  
# Layer 2 QoS. 802.1Q/VLAN Tag (VLAN classification for RTP)  
P51 = 0  
# Layer 2 QoS. 802.1p priority value (0 - 7)  
P87 = 0  
# No Key Entry Timeout. Default - 4 seconds.  
P85 = 4
```



```

# Use # as Dial Key (if set to Yes, "#" will function as the "(Re-)Dial" key). 0 - no, 1 - yes
P72 = 1
# Local RTP port (1024-65535, default 5004)
P39 = 5004
# Use Random Port. 0 - no, 1 - yes
P78 = 0
# Keep-alive interval (in seconds. default 20 seconds)
P84 = 20
# Use NAT IP. This will enable our SIP client to use this IP in the SIP message. Example 64.3.153.50.
P101 =
# STUN server
P76 = stun.mycompany.com

#-----
# Firmware Upgrade
#-----
# Firmware Upgrade. 0 - TFTP Upgrade, 1 - HTTP Upgrade.
P212 = {$firmware_upgrade_type}
# Firmware Server Path
P192 = {$firmware_srv_path}
# Config Server Path
P237 =
# Firmware File Prefix
P232 =
# Firmware File Postfix
P233 =
# Config File Prefix
P234 =
# Config File Postfix
P235 =
# Allow DHCP Option 66 to override server. 0 - No, 1 - Yes. Default is Yes.
# When set to Yes(1), it will override the configured provision path and method.
P145 = 1
# Automatic Upgrade. 0 - No, 1 - Yes (checking every defined days). Default is No.
P194 = {$auto_up}
# Check for new firmware every () minutes, unit is in minute, default is 7 days.
P193 = 10080
# Use firmware pre/postfix to determine if f/w is required
# 0 = Always Check for New Firmware
# 1 = Check New Firmware only when F/W pre/suffix changes

```

```

# 2 = Always Skip the Firmware Check
P238 = 0
# Authenticate Conf File
P240 = 0
# Off-hook Auto Dial
P71 =
# DTMF Payload Type
P79 = 101
# Onhook Threshold. Default 800ms.
# <value=0>Hookflash OFF
# <value=2>200 ms
# <value=4>400 ms
# <value=6>600 ms
# <value=8>800 ms
# <value=10>1000 ms
# <value=12>1200 ms
P245 = 8
# Syslog Server (name of the server, max length is 64 characters)
P207 =
# Syslog Level (Default setting is NONE)
# 0 - NONE, 1 - DEBUG, 2 - INFO, 3 - WARNING, 4 - ERROR
P208 = 0
# NTP Server
P30 = {$srvip}
# Allow DHCP Option 42 to override NTP server. 0 - No, 1 - Yes. Default is No.
# When set to Yes(1), it will override the configured NTP server.
P144 = 1
# Distinctive Ring Tone
# Use custom ring tone 1 if incoming caller ID is the following:
P105 =
# Use custom ring tone 2 if incoming caller ID is the following:
P106 =
# Use custom ring tone 3 if incoming caller ID is the following:
P107 =
# System Ring Tone
P345 = f1=440,f2=480,c=200/400;
### Call Progress Tones ###
# Syntax: f1=val,f2=val[,c=on1/off1[-on2/off2[-on3/off3]]];
# (Frequencies are in Hz and cadence on and off are in 10ms)
# Dial Tone

```

```

P343 = f1=350,f2=440;
# Message Waiting
P344 = f1=350,f2=440,c=10/10;
# Ring Back Tone
P346 = f1=440,f2=480,c=200/400;
# Call-Waiting Tone
P347 = f1=440,f2=440,c=25/525;
# Busy Tone
P348 = f1=480,f2=620,c=50/50;
# Reorder Tone
P349 = f1=480,f2=620,c=25/25;
# Disable Call Waiting. 0 - no, 1 - yes
P91 = 0
# Disable Direct IP calls 0 - no, 1 - yes
P1310 = 0
# Use Quick IP-call Mode. 0 - no, 1 - yes
P184 = 0
# Disable Conference. 0 - no, 1 - yes
P1311 = 0
# Lock Keypad Update. 0 - no, 1 - yes
P88 = 0
# Disable DND Button. 0 - no, 1 - yes
P1340 = 0
# Disable Transfer. 0 - no, 1 - yes
P1341 = 0
# Send Flash Event. 0 - no, 1 - yes
P74 = 0
# Headset TX gain (dB). 0 - 0 dB, 1 - -6 dB, 2 - +6 dB.
P1301 = 0
# Headset RX gain (dB). 0 - 0 dB, 1 - -6 dB, 2 - +6 dB.
P1302 = 0

#####
# Account Settings
#####
# Account Name
P270 = {$ext.line.1}
# SIP Server
P47 = {$srvip}
# Outbound Proxy

```

```

P48 = {$srvip}
# SIP User ID
P35 = {$ext.line.1}
# Authenticate ID
P36 = {$ext.line.1}
# Authenticate password
P34 = {$pass.line.1}
# Display Name (John Doe)
P3 = {$displayname.line.1}
# Use DNS SRV. 0 - No, 1 - Yes.
P103 = 0
# SIP User ID is phone number. 0 - no, 1 - yes
P63 = 0
# SIP Registration. 0 - no, 1 - yes
P31 = 1
# Unregister On Reboot. 0 - no, 1 - yes
P81 = 1
# Register Expiration (in minutes. default 1 hour, max 45 days)
P32 = 60
# Local SIP port (default 5060)
P40 = 5060
# SIP T1 Timeout. RFC 3261 T1 value (RTT estimate)
# 50 - 0.5 sec, 100 - 1 sec, 200 - 2 sec. Default 100.
P209 = 100
# SIP T2 Interval. RFC 3261 T2 value. The maximum retransmit interval for non-INVITE requests and INVITE
responses.
# 200 - 2 sec, 400 - 4 sec, 800 - 8 sec. Default 400.
P250 = 400
# SIP Transport. 1 - UDP, 2 - TCP.
P130 = 1
# Use RFC3581 Symmetric Routing
# 0 = No, 1 = Yes
P131 = 0
# NAT Traversal. 0 - yes, 1 - no, 2 - No, but send keep-alive
P52 = 1
# SUBSCRIBE for MWI. (Whether or not send SUBSCRIBE for Message Waiting Indication) 0 - No, 1 - Yes.
P99 = 0
# Proxy-Require (A SIP extension to enable firewall penetration)
P197 =
# Voice Mail UserID (User ID/extension for 3rd party voice mail system)

```

```

P33 = $ext
# Send DTMF. 8 - in audio, 1 - via RTP, 2 - via SIP INFO
# 11 - In Audio & RTP & SIP INFO, 9 - In Audio & RTP
# 10 - In Audio & SIP INFO, 3 - RTP & SIP INFO
P73 = 0
# Early Dial (use "Yes" only if proxy supports 484 response). 0 - no, 1 - yes
P29 = 0
# Dial Plan Prefix (dial plan prefix string).
P66 =
# Delayed Call Forward Wait Time. 1 to 120 seconds. Default 20 seconds.
P139 = 20
# Enable Call Features. 0 - no, 1 - yes
P191 = {$enable_call_features}
# Disable Call Log
# 0 = no, 1 = yes
P187 = 0
# Session Expiration (in seconds. default 180 seconds. Allowed value: 90-65535)
P260 = 180
# Minimum SE (in seconds. default 90 seconds, must be lower than or equal to P260)
P261 = 90
# Caller Request Timer (Request for timer when calling) 0 - no, 1 - yes
P262 = 0
# Callee Request Timer (Request for timer when called. i.e. if remote party supports timer but did not request for
one) 0 - no, 1 - yes
P263 = 0
# Force Timer (Still use timer when remote party does not support timer) 0 - no, 1 - yes
P264 = 0
# UAC Specify Refresher. 0 - omit, 1 - UAC, 2 - UAS
P266 = 0
# UAS Specify Refresher. 1 - UAC, 2 - UAS
P267 = 1
# Force INVITE (Always refresh with INVITE instead of UPDATE even when remote party supports UPDATE)
0 - no, 1 - yes
P265 = 0
# Enable 100rel. 0 - no, 1 - yes
P272 = 0
# Account Ring Tone. 0 - system ring tone, 1 - custom ring tone 1, 2 - custom ring tone 2
# 3 - custom ring tone 3.
P104 = 0
# Ring Timeout (in seconds. Between 30-3600, default is 60)
P1328 = 60

```

```
# Send Anonymous (caller ID will be blocked if set to Yes). 0 - no, 1 - yes
P65 = 0
# Anonymous Method. 0 - Use From Header, 1 - Use Privacy Header.
P268 = 0
# Auto Answer. 0 - no, 1 - yes
P90 = 0
# Allow Auto Answer by Call-Info. 0 - no, 1 - yes
P298 = 0
# Turn off speaker on remote disconnect. 0 - no, 1 - yes
P299 = 0
# Check SIP User ID for incoming INVITE: 0 - no, 1 - yes
P258 = 0
# Refer-To Use Target Contact
# 0 = no, 1 = yes
P135 = 0
# Disable Multiple Media Attribute in SDP. 0 - no, 1 - yes
P137 = 0
# Preferred Vocoder
# 0 - PCMU, 2 - G.726-32, 3 - GSM, 4 - G.723.1, 8 - PCMA,
# 9 - G.722, 18 - G.729A/B, 98 - iLBC
# First codec.
P57 = 0
# Second codec.
P58 = 8
# Third codec.
P59 = 4
# Forth codec.
P60 = 18
# Fifth codec.
P61 = 2
# Sixth codec.
P62 = 3
# Seventh codec.
P46 = 9
# Eighth codec.
P98 = 98
# SRTP Mode
# 0 = Disabled
# 1 = Enabled but not forced
# 2 = Enabled and forced
```

```

P183 = 0
# Special Feature. 100 - Standard, 101 - Nortel MCS, 102 - Broadsoft, 108 - CBCOM, 109 - RNK.
P198 = 100

#####
# End User Settings. Please do not edit this section.
#####
# End User Password
#P196 = 123
# DHCP support. 0 - yes, 1 - no
#P8 = 0
# PPPoE support. PPPoE user ID
#P82 =
# PPPoE password
#P83 =
# Host name, DHCP option 12. Max length allowed is 32 bytes.
P146 =
# Domain name, DHCP option 15. Max length allowed is 32 bytes.
P147 =
# Vendor Class ID, DHCP option 60. Max length allowed is 32 bytes.
P148 = Grandstream BT200
# Use this DNS server. (if specified).
#P92 = 192
#P93 = 168
#P94 = 0
#P95 = 2
# IP Address. Ignore if DHCP or PPPoE is used
#P9 = 192
#P10 = 168
#P11 = 0
#P12 = 1
# Subnet mask. Ignore if DHCP or PPPoE is used
#P13 = 255
#P14 = 255
#P15 = 255
#P16 = 0
# Router. Ignore if DHCP or PPPoE is used
#P17 = 192
#P18 = 168
#P19 = 1

```

```

#P20 = 1
# DNS 1. Ignore if DHCP or PPPoE is used
#P21 = 192
#P22 = 168
#P23 = 0
#P24 = 1
# DNS 2. Ignore if DHCP or PPPoE is used
#P25 = 0
#P26 = 0
#P27 = 0
#P28 = 0

#-----
# End User Time settings
#-----

# Time Zone. Offset in minutes to GMT
P64 = {$gmtoff}
# Allow DHCP Option 2 to override Time Zone setting. 0 - No, 1 - Yes.
# When set to Yes(1), it will override the configured Time Zone setting if available.
P143 = 1
# Daylight Savings Time. 0 - no, 1 - yes
P75 = 0
# Optional Rule
P246 = 3,2,7,2,0:11,1,7,2,0;60
# Date Display Format. 0 - Year-Month-Day, 1 - Month-Day-Year, 2 - Day-Month-Year
P102 = 1

# Display Account Name instead of Date. 0 - no, 1 - yes
P339 = 1
# Mute Speaker Ringer in Headset Mode. 0 - no, 1 - yes
P336 = 0

```